Roma Tre University
Ph.D. in Computer Science and Engineering

Root Cause Analysis and Forensics in Interdomain Routing: Models, Methodologies and Tools

Tiziana Refice

Root Cause Analysis and Forensics in Interdomain Routing: Models, Methodologies and Tools

A thesis presented by
Tiziana Refice
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Engineering

Roma Tre University
Dept. of Informatics and Automation

April 2009

Committee:
Prof. Giuseppe Di Battista

Reviewers:
Dott. Daniel Karrenberg
Prof. Giorgio Ventre

Contents

Contents

Introduction

The Internet is an interconnection of administrative domains called *Autonomous Systems* (*ASes*). Each AS contains one or multiple *destination networks* and each network is identified by an IP prefix. The *Border Gateway Protocol* (*BGP*) [RLH06] is the de-facto standard routing protocol used to exchange reachability information among ASes and a BGP session between two distinct ASes is called *peering*. Each AS learns through BGP its "best" *route* towards each destination in the Internet, updates it in response to *network events* (e.g., link failures, router resets, or policy changes) and propagates the change by BGP messages called *updates*. The propagation of BGP updates can be partially controlled via *routing policy* specifications.

In order to investigate the Internet behavior over time, several repositories provide historical data. Since 1997 and 1999, respectively, the University of Oregon *RouteViews Project* (*RV*) [roua] and the RIPE NCC *Routing Information Service* (*RIS*) [roub] spread worldwide passive *monitors* (or *vantage points*), which continuously gather BGP routing data from the Internet, permanently store them and make them publicly available. Currently, there are about 800 such monitors. Also, in 1995 the *Internet Routing Registry* (*IRR*) was established and started collecting inter-AS routing policies of many of the networks in the Internet with the main purpose to promote stability, consistency, and security of the global interdomain routing.

As the Internet becomes a more and more critical infrastructure, the need for understanding and (at least at some extent) controlling the interdomain routing increases. *Internet Service Providers* (*ISPs*) - in order to improve the quality of service offered to their customers - want to monitor the reachability of specific prefixes, check the effectiveness of their own routing policies, and assess the impact of traffic engineering configurations. In this context, it is crucial to be able to detect and debug misconfigurations or faults, in order to possibly fix them. More generally, the problem of identifying Internet events, locating

their root causes, and understanding their dynamics is attracting increasing attention from both researchers and network operators.

However, despite the large amount of research effort, routing dynamics diagnosis remains very difficult for several reasons: (*i*) The system has a sheer size. As of December 2008, there are about 280,000 prefixes and more than 30,000 Autonomous Systems densely connected between each other. (*ii*) The Internet is highly dynamic. In fact, RIS' and RV's monitors currently receive an average of about 1,500 BGP updates per minute, with peaks of more than 50,000 updates per minute. (*iii*) Due to complex interconnects among ASes and routing policies, the effects of network events are often separated (both in time and space) from their causes and different vantage points record different data in response to the same routing changes. Also, multiple routing events can occur simultaneously. Overall, given such size and dynamics, "naive" approaches to extract relevant information from the Internet routing data are neither effective nor efficient.

Therefore, both researchers and network operators interested in understanding the interdomain routing have to cope with several major challenges. First, in order to deal with such a huge and complex network, they need to define what to measure, i.e., they need a model of the Internet routing that captures the main dynamics, filtering out the "noise" (e.g., routing changes that do not provide information relevant to the identification of network events). Based on such model, they need a methodology that, given the currently available data sources, detects network events and infers when and where they happened. Furthermore, they need tools that efficiently handle the huge amount of data, support the analysis of the network behavior over time, and provide real-time information in order to spot and possibly fix outages as soon as they occur. Since the analysis of network events often requires manual work, effective paradigms for the visualization of routing data are also very helpful. Previous works leave most of these problems still open.

The research work described throughout this thesis addresses these problems and proposes approaches to (at least partially) solve them. Namely, this thesis presents the following contributions.

Chapter 2 illustrates a new perspective to drive the analysis of the Internet dynamics without getting lost in the huge BGP dataset. Basically, while previous works usually address the root cause analysis from a "global perspective" - i.e., by taking into account the dynamics of the whole Internet and trying to identify major events affecting it - Chapter 2 tackles the same problem with an ISP-oriented approach: it assumes that ISPs are usually more interested in the reachability of their own prefixes, rather than in the status

of the whole Network; hence, it focuses the analysis on user-specified prefixes and correlates their behaviors to the global Internet dynamics. In particular, Chapter 2 formally models the Internet as a flow-based system, where monitors are the sources of the flows and ASes originating BGP updates are the sinks. Chapter 2 also defines a methodology which correlates such flow variations to routing changes in order to spot network events and the root causes that triggered them. Furthermore, BGPath has been developed to support this methodology and Chapter 2 describes its main features. BGPath is a publicly available tool that uses BGP data collected by the RIS and the RV projects and provides the user with routing information from both a single and cross-vantage point views. BGPath also assesses the reliability of the collection system, in order to avoid measurement artifacts. The algorithms BGPath relies on are shown to efficiently process huge streams of BGP data, fulfilling nearly-real time constraints.

While the ISP-oriented approach presented in Chapter 2 gives a good insight on both major and minor events affecting specific portions of the Internet, approaching the root cause analysis problem from a "global perspective" usually does not provide with such fine-grained results. On the other hand, the global approach is critical to identify major interdomain events, without any a-priori knowledge of the prefixes and/or the ASes involved. This thesis explores this perspective too. Specifically, Chapter 3 proposes a novel methodology based on the *Principal Component Analysis* (*PCA*), a well-known statistical technique that is commonly used to reduce the number of dimensions of multi-dimensional datasets in order to highlight the most significant trends of the data. Since the interdomain routing dataset is inherently multi-dimensional (in time, space, prefixes, observation points, ...), Chapter 3 suggests to apply the PCA to this dataset in order to identify the most significant contributors to the Internet dynamics.

BGP data collected by RIS' and RV's monitors provide a detailed view of the actual status of the interdomain routing. However, it does not report all the inter-AS peering relationships which are not active. For example, in "normal" conditions, backup links do not appear in the routing tables. Still, in order to understand the reasons behind some network events and to predict the evolution of the routing when an event occurs, such information is actually very important. To cope with the intrinsic limitations of the RIS and RV dataset, Chapter 4 analyzes the data stored in the Internet Routing Registry and describes how to extract peering relationships from routing policies collected within. Moreover, the proposed approach specifies how to solve inconsistencies among the distinct databases the IRR consists of. The obtained

results show that - even though the IRR data is often out-of-date, it still provides a quite unique amount of topological information which usually does not appear in the global routing.

The research work described in the previous chapters relies on the assumption that Internet is a graph where ASes are atomic entities in the interdomain routing. However, recent papers [MFM+06, MUF+07] show that such a model can mislead the understanding of the global routing behavior. Thus, Chapter 5 investigates this problem by measuring the route diversity that can be observed by passive remote vantage points, defining a methodology to compute it from a dynamic BGP dataset and characterizing it in terms of location of ASes in the Internet customer-provider hierarchy and choice of monitors.

The thesis ends with Chapters 6,7, which document forensic analysis of two well-know events that occurred at the beginning of 2007, where models, methodologies and tools described in the previous chapters are exemplified using real case studies.

Chapter 1

Background

1.1 Internet and Interdomain Routing

The Internet is divided into administrative domains called *Autonomous Systems* (*AS*), each adopting consistent routing policies. An AS is identified by a number called *Autonomous System Number* (*ASN*). Every AS contains one or multiple *destination networks*. Each destination network is represented by an IP address prefix. As of December 2008, there are about 30,000 Autonomous Systems and more than 250,000 prefixes.

The *Border Gateway Protocol* (*BGP*) [RL95, RLH06] is the routing protocol used to exchange reachability information between ASes. Two ASes that exchange routing information using BGP are said to have a *peering* between them. The ASes having a peering with an AS A are termed *peers* (or *neighbors*) of A. A BGP router stores in its *Routing Information Base* (*RIB*) the *prefixes* it can reach, and for each of them an *AS-path*. An AS-path is the sequence of ASes used to reach the destination prefix. Routes are propagated by BGP messages called *updates*. BGP is an incremental protocol: once two BGP routers establish a peering, they exchange their whole RIB each other; this process is called *table transfer*. Further updates are sent only if a route changes, in response to *network events* (e.g., link failure, router reset, or policy change). Once a BGP router receives from any of its *peers* an update for a prefix p, it recomputes its best path towards p, possibly changes its own RIB and propagates the update to its peers.

Routing policies can be configured to decide which neighbors to send routes to and receive routes and best path selection is based on inter-AS *customer-*

provider relationships. Such relationships define a *hierarchy* of all the ASes.

1.2 Interdomain Routing Data Sources

1.2.1 Actual Routing Data

To obtain information about the evolution of the Internet routing state, projects such as the RIPE NCC's *Routing Information Service* (*RIS*) [roub] and the University of Oregon's *RouteViews Project* (*RV*) [roua], spread around the world several passive collection boxes, called *(Remote) Route Collectors* (*RRCs*). Each route collector peers with several BGP routers, called *Collector Peers* (*CPs*) or *monitors*, belonging to various ASes. The routing tables of all RRCs and the updates they receive are periodically dumped, permanently stored, and made publicly available. Some collector peers provide information about all the prefixes on the Internet, while others only provide information about a subset of them. We call the former *full collector peers*, the latter *partial collector peers*.

1.2.2 Internet Routing Registry

The *Internet Routing Registry* (*IRR*) [ripc, irra] is a large distributed repository of information, containing the inter-domain routing policies of many of the networks that compose the Internet. The IRR was established in 1995 with the main purpose to promote stability, consistency, and security of the global Internet routing. The IRR can be used by operators to look up peering agreements, to study optimal policies, and to (possibly automatically) configure routers.

The IRR consists of several databases, called (*routing*) *registries*. Some routing registries are maintained by *Regional Internet Registry* (e.g., RIPE [regc], ARIN [rega]) and contain information over wide geographic regions, while others are maintained by *Local Internet Registries* (e.g., VERIO [regd], LEVEL3 [regb]) and describe routing policies of the customers of a specific Internet Service Provider.

The registration and maintenance of routing policies are performed on a voluntary basis by network operators, who may register such policies at one or more registries. As a consequence, information in the IRR may be incorrect, incomplete, or outdated. Indeed, some large ISPs and Internet Exchange Points rely on the IRR for route filtering and their customers are required to document their policies in a registry.

1.2. Interdomain Routing Data Sources

The Routing Policy Specification Language

The routing policies stored in the IRR are described using the *Routing Policy Specification Language* (*RPSL*) [AVG+99, MSO+99] or its more recent variant *RPSLng* [BDPR05], which introduces support to both multicast and IPv6. RPSL is an object-oriented language that defines 13 classes of objects. Routing policies are described in the `import`, `export`, and `default` attributes of `aut-num` objects. In turn, `aut-num`s may reference other objects that contribute to the specification of the policies, such as `as-set`s and `peering-set`s.

What follows is a portion of an RPSL `aut-num` object from the RIPE registry which describes the inter-domain policies of AS137 (last updated 06/11/07). The portion of the `import` (`export`) attribute following the `from` (`to`) keyword is a very simple example of *peering specification*. The object indicates that AS137 accepts any route sent to it by AS20965 and by AS1299 and propagates to AS1299 all the routes originated by ASes belonging to the `as-set` named AS-GARR (an `as-set` is an RPSL object that specifies a set of ASes). This implies that AS137 has a peering with AS20965 and AS1299.

```
aut-num: AS137
import: from AS20965 action pref=100;
      from AS1299 action pref=100;
      accept ANY
[...]
export: to AS1299 announce AS-GARR
[...]
changed: vincenzo.puglia@garr.it 20070611
source: RIPE
```

Peval and IRRd

Peval is a policy evaluation tool conceived to write router configuration generators and it is part of the *Internet Routing Registry Toolset* (*IRRToolSet*) [irrc] suite. Peval takes as input an RPSL expression and evaluates it by applying RPSL set operators (`AND`, `OR`, `NOT`) and by expanding `as-set`s, `route-set`s, and AS numbers into the corresponding sets of prefixes. Alternatively, Peval can stop the expansion at the level of ASes. The IRR data can also be accessed through the *Internet Routing Registry Daemon* (*IRRd*) [irrb], a freely available stand-alone IRR database server supporting both RPSL and RPSLng.

1.3 Principal Component Analysis

Principal Components Analysis (*PCA*) is a well-known statistical technique used for understanding the variance of a given data set. PCA maps a set of points from a n-dimensional space into a new orthogonal n-dimensional space, where the variance of the original data along each axis is maximized. The axes of the new space are called *principal components*. The coefficients of the new reference axes are called *loadings*, and the projections of the original data onto these axes are called *scores*.

Given a $m \times n$ matrix $\mathbf{X}$, where each row represents a point, PCA computes n principal components $\mathbf{v}_i{}_{i=1}^{n}$ defined as follows: $\mathbf{v}_k = argmax_{||\mathbf{v}||=1}||(\mathbf{X}^T - \sum_{i=1}^{k-1} \mathbf{X}^T \mathbf{v}_i \mathbf{v}_j^T)\mathbf{v}||$. The principal components are the n eigenvectors of the estimated covariance matrix and are ranked according to the amount of variance they capture in the original data. The variance of each component is described by the corresponding eigenvalue. When the input matrix is zero-mean, the first principal component contains the most variance in the original data, and any other k^{th} principal component - with $k = 2, ...n$ - identifies the maximum variance in the remaining data, i.e. the original data after removing the contributions of the previous $k - 1$ components.

Typically, the first few principal components capture almost all the variance in the input dataset. Thus, PCA is usually applied for dimensionality-reduction of datasets to obtain more compact representations in lower dimensional subspaces, by keeping lower-order components and ignoring higher-order ones.

Part I

Models, Methodologies and Tools

Chapter 2

Detecting and Analyzing Inter-domain Events

Interdomain routes change over time, and it is impressive to observe up to which extent. Routes, even the most stable, can change many times in the same day and sometimes in the same hour or minute. Such variations can be caused by several types of events, e.g., the change of the routing policies of an ISP, the reboot of a router, or the fault of a link. Some events are physiological to the network, while others are anomalous.

In this chapter we do a step towards the identification of the cause of route changes, a problem that is attracting increasing attention from both researchers and network administrators. Namely, we propose a methodology for analyzing a given BGP route change in order to, at least partially, locate the event that triggered the change. The methodology is supported by an on-line service.

The main results presented in this chapter are also described in [CCD+08, CRC+08].

This chapter is organized as follows. Section 2.1 describes previous work and our contributions to this research area. Sections 2.2 and 2.4 describe, respectively, the flow-based model and the methodology we defined. Our dataset is described in Section 2.3. Section 2.5 describes the BGPath́s user interface through a real usage scenario and provides a high level description of how the BGP data is processed. The underline algorithms are detailed in Section 2.6. The effectiveness of the methodology and the tool is discussed in Section 2.7 by means of simulation experiments and real world data analysis. Section 2.8 concludes this chapter.

2.1 Introduction

We consider the following scenario. A network administrator of an Internet Service Provider observes that one of the prefixes announced by its Autonomous System to the Internet had a BGP path change at a certain time. For example, prefix p announced by AS1 usually reaches AS4 passing through AS2 and AS3, while suddenly it started using a different path through AS5 and AS6. The network administrator would like to know why that change happened.

Previous Work

Recent works (e.g., [WMW+06]) underline the impact of routing changes in end-to-end performance. Also, this issue becomes much more important as services requiring almost constant delay, limited jitter and packet loss, gain popularity. Hence, many ISPs are interested in understanding what happens to their prefixes in the interdomain routing.

Actually, many research works studied BGP routing dynamics in the last few years. Their contributions can be broadly classified as follows. There are black-box approaches, that apply statistical techniques to group BGP updates into sets that are supposed to be triggered by the same underlying event. Ref. [XCZ05] uses the Principal Components Analysis, [ZYZ+04] uses statistics-based anomaly detection, and [ZRF05] exploits the wavelet transform. Other authors propose white-box approaches. In [FMM+04, CGH03, CSK03] streams of BGP updates are analyzed, correlating information across time, topology, collectors, and prefixes. Ref. [LNMZ04] describes an algorithm, that pinpoints the origin of routing changes due to a link failure or a link restoration, assuming shortest path routing. Finally, some authors (see, e.g., [WMRW05]) propose to add an infrastructure to the Internet in order to monitor route changes.

Those contributions generally aim at reporting a full set of events that happened in the network in a given time slice. Roughly, updates are first grouped into clusters, and then events are detected by analyzing multiple clusters. In this modus operandi, the correlation between an update and an event can be biased by the a priori generation of the clusters.

Our Contributions

Taking into account the scenario described at the beginning of this section, we propose to tackle the problem from a different perspective. We assume the perspective of an ISP, that is not interested in what happens to the network in

general but is rather interested in what happens at a certain time to (some of) its own prefixes. Hence, instead of analyzing a bulk of updates for detecting events in the network, we analyze a specific BGP-update trying to locate its originating event.

In this chapter we present the following contributions. In Section 2.2, we show that BGP updates have a flow-based behavior, where the term "flow" is used with its graph-theoretic meaning. The collectors of updates are sources of flow and the ASes originating prefixes are sinks. Exploiting this property, we propose a flow-based model of BGP updates. As far as we know, this is the first time that BGP updates are modeled in terms of a flow system. As a side effect, we put in a flow-based perspective the concept of link-rank, defined in [LMZ04]. Further, this section introduces the new concept of global-rank. In Section 2.4, we propose a methodology for analyzing a given BGP route change c in order to, at least partially, identify and locate the event that triggered c. The cornerstones of the methodology are: (i) A data quality analysis for discarding unreliable data, extending the approach of [ZKL+05]. (ii) A macro-events detection analysis, focused on local and global ranks. (iii) A fine-grained analysis that analyzes flow changes in a relevant part of the network. The methodology is illustrated by several examples from a reference week. The effectiveness of the methodology is discussed in Section 2.7 by means of simulation experiments and real world data analysis. Our data sources are described in Section 2.3.

The methodology described in Section 2.4 requires the analysis of huge amount of data, and hence it would be unfeasible if not supported by some automatic facility. We developed an on-line service that offers many tools to support the methodology. A prototype version is available at
`http://nerodavola.dia.uniroma3.it/rca/`

2.2 Flow-based Model of Inter-domain Routing Dynamics

Several models have been proposed to study the evolution of interdomain routing. Most of them assume that each AS can be collapsed into a single router, while others [MFM+06] represent the internal structure of each AS with different levels of accuracy. The first approach can be too coarse-grained to capture the impact of the internal routing of an AS on the evolution of the Internet. On the other hand, the second approach contrasts with the fact that the currently available methodologies and data are not able to provide a fine-grained

complete and accurate description of the internal structure of an AS.

In this section, we introduce a model based on the concept of flow. The model is shown to be valid not depending on the internal structures of any AS. The validity of the model has the benefit of allowing correct deductions in Root Cause Analysis of interdomain routing. Of course, it also has the drawback of not capturing dynamics internal to an AS.

Basic Terminology

We consider the following sets. $\mathcal{ASes}$ is the set of all the known ASes, $\mathcal{ASes} = \{1, \ldots, 65535\}$. Since we will consider a graph whose nodes are the elements of $\mathcal{ASes}$, the ASes will also be called *vertices*. $\mathcal{T}$ is the set of all the considered instants of time when a BGP update is received by a RRC from a collector peer. $\mathcal{CP}$ is the set of all the collector peer identifiers.

An *AS-path* (or simply *path*) $\pi = (as_n, \ldots, as_0)$ is a sequence of ASes, where $as_i \in \mathcal{ASes}$. as_0 is called *origin*. The empty path is denoted by ϕ. $\mathcal{AP}$ is the set of all known AS-paths. An *edge* is a pair (as_{i+1}, as_i) of ASes that are adjacent in some AS-path. We consider edges as directed, i.e. $(v, w) \neq (w, v)$. We say that a path *contains* an edge, $\pi \supseteq (as_{i+1}, as_i)$. Each collector peer *cp* stores in its RIB the AS-paths it selects to reach all the observed prefixes.

An *update* u is a quadruple (cp, p, π, t) where $u.cp \in \mathcal{CP}$ is the CP that collected the update, $u.p \in \mathcal{P}$ is the prefix contained in the update, $u.\pi \in \mathcal{AP}$ is the AS-path announced by the update, and $u.t \in \mathcal{T}$ is the time when the update has been collected. If $u.\pi \neq \phi$ then u is an *announcement*, otherwise it is a *withdrawal*. $\mathcal{U}$ is the set of all known updates. The last update u that collector peer *cp* received for prefix p before time t is denoted $\ell_{cp}(p, t)$. Formally, $\ell_{cp}(p, t)$ is such that $\ell_{cp}(p, t).t < t$ and $\nexists u \in \mathcal{U} \mid u.cp = cp \ \wedge \ u.p = p \ \wedge \ \ell_{cp}(p, t).t < u.t < t$.

A *route change* (or simply *change*) occurs every time a collector peer *cp* updates its route to a prefix p. Formally, a change c triggered by an update u is a quintuple $(cp, p, \pi_{old}, \pi_{new}, t)$, where π_{old} is the path to prefix $c.p$ that $c.cp$ uses before $c.t$ ($\pi_{old} = \ell_{cp}(p, t).\pi$) and π_{new} is the path that $c.cp$ uses after $c.t$ ($\pi_{new} = u.\pi$). If both $c.\pi_{old} \neq \phi$ and $c.\pi_{old} \neq c.\pi_{new}$, the route change is also called *path change*.

Global, Local and Origin Ranks

We now define three concepts that will be crucial for the methodology described in Section 2.4, called *local rank*, *global rank*, and *origin rank*. While the first

has been introduced in [LMZ04], the others are, as far as we know, unexplored concepts.

Given a collector peer cp, the *local rank* of an edge e at time t - denoted by $lrank(cp, e, t)$ - is defined as the number of prefixes whose path at time t, as observed by cp, contains e. Observe that, since cp's RIB can change over time, the local rank depends on time (as well as on cp and e). Formally, $lrank(cp, e, t) = |P_{cp}(e, t)|$, where $P_{cp}(e, t) = \{p \in \mathcal{P} \mid e \subseteq \ell_{cp}(p, t).\pi \vee \exists u = (cp, p, \pi, t) \in \mathcal{U} \mid u.cp = cp, u.t = t, e \subseteq u.\pi\}$.

Given a set of collector peers $\mathcal{CP}$, the *global rank* of edge e at time t - denoted by $grank(e, t)$ - is defined as the number of distinct prefixes p for which there exists at least one $cp \in \mathcal{CP}$ such that its AS-path towards p at time t contains e. Also the global rank of an edge changes over time. Formally,

$$grank(e, t) = |P(e, t)|, P(e, t) = \bigcup_{cp \in \mathcal{CP}} P_{cp}(e, t).$$

Intuitively, while the local rank measures the number of prefixes that are observed passing through an edge by a single CP, the global rank measures the number of distinct prefixes that are observed passing through an edge by any CP $\in \mathcal{CP}$. We stress that, since the global rank takes into account all the collector peers at the same time, it provides a way to analyze interdomain routing from a cross-vantage point perspective.

Fig. 2.1 illustrates the values of local and global ranks of the edges of a fragment of Internet at a certain time t. For example, the label $(1, 2, 2, 3)$ on edge $(as1, as2)$ states that $lrank(cp_1, (as1, as2), t) = 1$ since cp_1 sees just the green dashed prefix traversing $(as1, as2)$. Also, $grank((as1, as2), t) = 3$ since $(as1, as2)$ is traversed by all three prefixes. Note that $grank((as1, as2), t) \neq \sum_{i=1,2,3} lrank(cp_i, (as1, as2), t)$. Observe that, even if cp_2 and as_2 have multiple peerings, according to our definitions, we labelled their pair only once.

Since the local rank relies on a single vantage point cp, this metric can be biased by malfunctions of cp, while the global rank is more resilient to faults affecting only a small subset of all collector peers. Moreover, network-related issues (such as congestion or large scale attacks), which can impact the monitoring system as shown in [WZP+02], are less likely to affect the global rank metric, given the wide geographical distribution of the collector peers. Summarizing, the global rank allows us to cope with what can be broadly classified as "noise" within the collection system. On the other hand, the local rank provides fine-grained information which is valuable to analyze a network event which only involves a relatively small portion of the Internet. In this

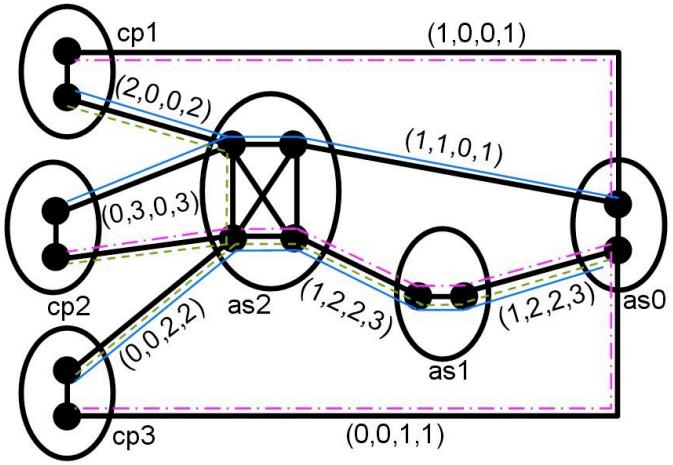


Figure 2.1: Big points represent routers, thick solid black lines represent IBGP or EBGP peerings between routers, and ellipses represent ASes. ASes cp_i, $i \in \{1,2,3\}$, contain collector peers. Each edge e inter-AS is labelled with a quadruple containing $lrank(cp_1, e, t)$, $lrank(cp_2, e, t)$, $lrank(cp_3, e, t)$, and $grank(e, t)$. AS as_0 originates three prefixes. Thin blue solid, green dashed, and pink mixed lines represent the routes to such prefixes observed by collector peers.

case, in fact, the evolution of *grank* could be misled by collector peers which do not observe the event. Overall, local and global ranks provide complementary benefits and partially compensate each other's weaknesses.

Finally, given a collector peer cp we define the *origin rank* of AS v at time t as $\theta(cp, v, t) = |P(cp, v, t)|$, where $P(cp, v, t) = \{p \in \mathcal{P} \mid \ell_{cp}(p, t).\pi = (as_n, ..., v), n \geq 0 \ \vee \ \exists u \in \mathcal{U} \ \mid \ u.cp = cp, u.t = t, u.\pi = (as_n, ..., v)\}$. Notice that function $\theta(cp, v, t)$ represents the number of prefixes that, at time t, are known by cp as originated by AS v. As an example, consider collector peer cp_1 in Fig. 2.1. $\theta(cp_1, as_0, t) = 3$ and $\theta(cp_1, as, t) = 0 \ \forall as \neq as_0$.

We denote by $\overline{lrank}$ ($\overline{grank}$) the weighted average of the local (global) rank of an edge over time.

Flows of Prefixes

Whenever a negative (positive) variation of a local rank is observed during a given time interval, it is interesting to further investigate where prefixes "went

to" ("came from"). Intuitively, prefixes move around on the AS graph, as well as water would move in a pipe network. This analogy introduces the concept of flows of prefixes. Tracking the flow of prefixes along different paths can be done by adapting the well-known concept of flow system to the interdomain routing.

Given a directed graph $G = (V, E)$, a specific vertex as_n called *source*, and a mapping between vertices and flow absorption $g : V \to \mathbb{Z}$, then a *flow system* is a function $f : E \to \mathbb{Z}$ where $\forall v \in V, v \neq as_n$,

$$\sum_{(u,v)\in E} f(u,v) - \sum_{(v,w)\in E} f(v,w) = g(v).$$

Theorem 1 shows that functions $lrank(cp, e, t)$ and $\theta(cp, v, t)$ define a flow system at time t. Intuitively, we have that the source of the flow is the AS as_n in which cp is located, and the sinks are all the ASes that originate some prefixes, as observed by cp. A prefix contributing to one unit of the local rank of some edge e contributes also to one unit of the amount of flow traversing e. As an example, consider collector peer cp_1 (in as_n) of Fig. 2.1. For instance, for as_2, we have that the sum of the local ranks over incoming edges $(cp_1, as2)$ is 2, and the sum of the local ranks over outgoing edges (as_2, as_0) and (as_2, as_1) is 2. On the other hand, since as_2 does not originate any prefix known to cp_1, $\theta(cp_1, as_2, t) = 0$. Hence, the flow around as_2 is conserved.

Theorem 1. *At a specific time instant, functions lrank and θ define a flow system.*

Proof. Select a specific instant t and a specific collector peer cp. Consider the value $x = \theta(cp, v, t)$ of function θ for any vertex v. Because of the definition of θ we have that for each unit of flow in x there exists a prefix p such that either $p \in \mathcal{P} \mid \ell_{cp}(p,t).\pi = (as_n, ..., v), n \geq 0$ or $\exists u \in \mathcal{U} \mid u.cp = cp, u.t = t, u.\pi = (as_n, ..., v)$. In both cases we identify an update u that is received from as_n and originates from v. Consider a sequence of two consecutive edges (as_{i+1}, as_i) and (as_i, as_{i-1}) contained in $u.\pi$, u contributes with one unit of flow both to $lrank(cp, (as_{i+1}, as_i), t)$ and to
$lrank(cp, (as_i, as_{i-1}), t)$. Hence, for each AS $as_i \neq v$, u does not affect the balance of as_i. This means that for each vertex v, if we consider only paths not ending with v we have

$$\sum_{w\in V} lrank(cp, (w, v), t) = \sum_{w\in V} lrank(cp, (v, w), t).$$

Now, each path ending with v increases both the flow on an incoming edge, $lrank(cp,(w,v),t)$, and $\theta(cp,v,t)$. Then we conclude that, $\forall v \in V$,

$$\sum_{w \in V} lrank(cp,(w,v),t) = \sum_{w \in V} lrank(cp,(v,w),t) + \theta(cp,v,t).$$

□

Observe that Theorem 1 holds even if some ASes perform BGP prefix aggregation. In fact, in this case a collector peer is unable to track all the prefixes contained in the aggregation and the aggregated prefix counts for just one unit of flow.

We stress that functions *grank* and θ do not define a flow system. As a counterexample, consider again AS as_2 in Fig. 2.1. We have that the algebraic sum of the global ranks of the edges incident on as_2 is not zero.

Theorem 1 is useful to depict a snapshot of the network at a given instant, while in Theorem 2 we relate the flows of two different instants of time. We define the functions

$$\Delta lrank_t^{t+\tau}(cp,e) = lrank(cp,e,t+\tau) - lrank(cp,e,t)$$

that captures local rank variations (flow variations) between t and $t+\tau$, and the function

$$\Delta\theta_t^{t+\tau}(cp,v) = \theta(cp,v,t+\tau) - \theta(cp,v,t).$$

that accounts for the variation in the number of prefixes that are known by *cp* as originated by v.

Theorem 2. *Functions* $\Delta lrank_t^{t+\tau}(cp,(v,w))$ *and* $\Delta\theta_t^{t+\tau}(cp,v)$ *define a flow system.*

Proof. For the sake of simplicity, we use $l((v,w),t)$ in substitution of $lrank(cp,(v,w),t)$. $\forall v \in V$:

$$\begin{aligned}
&\sum_{w \in V} \Delta lrank_t^{t+\tau}(cp,(w,v)) - \sum_{w \in V} \Delta lrank_t^{t+\tau}(cp,(v,w)) = \\
&\sum_{w \in V} l((w,v),t+\tau) - \sum_{w \in V} l((v,w),t+\tau) + \\
&- \sum_{w \in V} l((w,v),t) + \sum_{w \in V} l((v,w),t) = \\
&\theta(cp,v,t+\tau) - \theta(cp,v,t) = \Delta\theta_t^{t+\tau}(cp,v).
\end{aligned}$$

□

Observe that, because of the high connectivity of the Internet, a collector peer is likely to be able to reach a constant number of prefixes over time. Also, each of such prefixes is typically announced always by the same origin. Hence, we expect that function $\Delta\theta_t^{t+\tau}(cp, v)$ is zero in most cases. That is, we expect that the flow is overall conserved over time.

2.3 Our Dataset

Our work (and BGPath) relies on BGP data obtained from both RIS [roub] and RV [roua]. Overall, these projects (as of February 2007) provide 526 collector peers, 30% of which are full collector peers.

Examples and statistics presented in this chapter refer to the data collected from 12/26/2006 to 01/02/2007. We chose this time interval, referred to as *reference week*, because it featured massive BGP activity due to Taiwan earthquakes and it preceded the fix of a bug affecting RIS collectors. The reference week contains $320,678,893$ updates ($\sim$ 46M updates/day on average) with $7,537,378$ distinct paths on $70,078$ distinct peerings and $24,493$ distinct ASes. The number of observed prefixes is $235,725$.

2.3.1 Reliability Screening

In order to discard data coming from faulty or misconfigured collector peers, we periodically perform a data cleaning step, called *Reliability Screening*. The unreliability of a collector peer can be due to several reasons, including bugs in routing or collection software (see [Kon03] for details), major asynchronies between the collector peer and its route collector, and poor standard compliance (e.g., some vendors do not implement highly recommended optional timers).

The screening of a collector peer cp over a time interval $[t_{start}, t_{end}]$ is executed as follows: (*i*) we make a local copy of the RIB of cp at t_{start}, (*ii*) we modify the copy according to the updates collected by cp during $[t_{start}, t_{end}]$, (*iii*) we compare the modified copy to the RIB dumped by cp at t_{end}, (*iv*) we decide if cp is reliable evaluating the ratio between number of mismatches and its average RIB size.

Reliability Screenings performed during several experiments led to the detection of a major problem that affected RIS route collectors since May 2005. Overall, the problem affected 44 collector peers, 12 of which were full collector peers. Contacting the RIS maintainers resulted in fixing this problem by January 2$^{\text{nd}}$, 2007.

2.4 Methodology to Detect and Analyze Inter-domain Events

We present a methodology for analyzing a given route change c within the model of Section 2.2. The goal is to identify the portion of the Internet where the event that caused c happened. The methodology consists of three steps. *Collector Peer Check and Selection*: We check the availability of collector peers, and we select a set of collector peers that will be considered in the following analysis. *Macro-Events Detection*: We look for patterns of macro-events, by exploiting the global and local ranks of some edges. This step relies on Theorems 1 and 2. *Fine-Grained Analysis*: If no macro-event has been detected in the previous step, we perform a fine-grained analysis based on several patterns that are consequences of Theorem 2.

Before starting the description of the steps, we underline an issue related to the timing of network events. In several points of the methodology, we analyze what happens in a time interval including the time $c.t$ of the input route change. According to [LWVA01], we consider the time interval $[c.t - \Delta, c.t + \Delta]$, with Δ =180 seconds, as a reasonable compromise between accuracy and feasibility and we refer to it as $T_{c.t}$. However, the methodology does not depend on this choice.

2.4.1 Collector Peer Check and Selection

Before starting the analysis of the route change c, the methodology requires to execute the Reliability Screening (Section 2.3.1) in order to discard all the unreliable CPs.

Also, collector peers may reboot. If the collector peer $c.cp$ that receives c has a reboot in $T_{c.t}$, we interrupt the analysis because the data collected through $c.cp$ may be too noisy. Moreover, in the analysis of c, we will rely not only on $c.cp$, but also on other collector peers. Hence, in this step we look for all the collector peers that had a reboot in $T_{c.t}$. Information extracted from those collector peers is not further considered. We detect a reboot by either analyzing BGP session state messages, when available, or by seeking for table transfers using the algorithm described in Section 2.6.1.

Among all reliable collector peers without any reboot in $T_{c.t}$, we select those that belong to the ASes of the paths $c.\pi_{old}$ and $c.\pi_{new}$, because they are the most relevant for the subsequent analysis since they provide the closest perspective to analyze c.

2.4.2 Macro-Events Detection

In this step we try to relate c to a macro-event by performing first a global rank analysis and then a local rank analysis. We regard as *macro-events* those which affect either the physical or the logical network topology (e.g. an interdomain link fault/restoration, a BGP router fault/restoration, or a BGP session shutdown/setup).

The evolution of the global rank $grank(e, t')$ with $t' \in T_{c.t}$ is considered for each edge e in $c.\pi_{old}$ and $c.\pi_{new}$. Namely, we check if some edge e in $c.\pi_{old}$ or in $c.\pi_{new}$ has a relevant global rank variation and has a value near to zero in $T_{c.t}$. This occurs when no collector peers see any prefix passing through e, and it is a reasonable evidence that e is involved in some way in the event that caused c. We identified three patterns of global rank evolution: (p1) a sudden loss of all prefixes, (p2) a sudden gain of new prefixes starting from 0 prefixes, or (p3) a sudden loss (gain) followed by the resume of the previous situation. Each patter possibly refers to different types of macro-events. E.g. (p1) describes an interdomain link e that fails and loses connectivity to all the prefixes. Once fixed, prefixes might be routed through e again (p2). According to our experience, both gains and losses usually occur within short time periods, due to BGP convergence time (see [LWVA01]). We relate macro-events to the evolution of the global rank of an edge e because it provides a global perspective given by the simultaneous views of e from several collector peers. As the number of collector peers that can see e decreases, this global perspective is more biased. In order to cope with this behavior, we define the *rank diversity*. The rank diversity of e is a pair $\langle n, \sigma_x/\overline{x} \rangle$, where n is the number of collector peers cp having $\overline{lrank}(cp, e) > 0$, σ_x and $\overline{x}$ are the standard deviation and the average, respectively, of such $\overline{lrank}(cp, e)$. We say that the rank diversity is *high* if n is large and $\sigma_x/\overline{x}$ is small. In fact, if n is large we have many collector peers that can see e, and when $\sigma_x/\overline{x}$ is small we have that the collector peers see a similar number of prefixes through e. The global rank analysis provides more valuable information on edges having higher rank diversity.

As an example, we analyze the path change c affecting the prefix $c.p =$ 202.41.242.0/24, with $c.\pi_{old} =$ (2497, 4134, 4847, 37942) and $c.\pi_{new} =$ (2497, 2914, 4134, 4847, 37942), observed by $c.cp =$ 198.32.176.24 at time $c.t =$ UTC 30/12/06 05:52:24. First, we check collector peers availability. We identify $\sim$ 20 collector peer resets in $T_{c.t}$ and discard data coming from these CPs. Then, we evaluate the global rank of the edges in $c.\pi_{old}$ and $c.\pi_{new}$. We have that $grank(e) = 0$ in $T_{c.t}$, with $e = (2497, 4134)$. Since $\overline{grank}(e) = 3,166$, edge e has a significant rank variation (Fig. 2.2.a). The rank diversity of e is

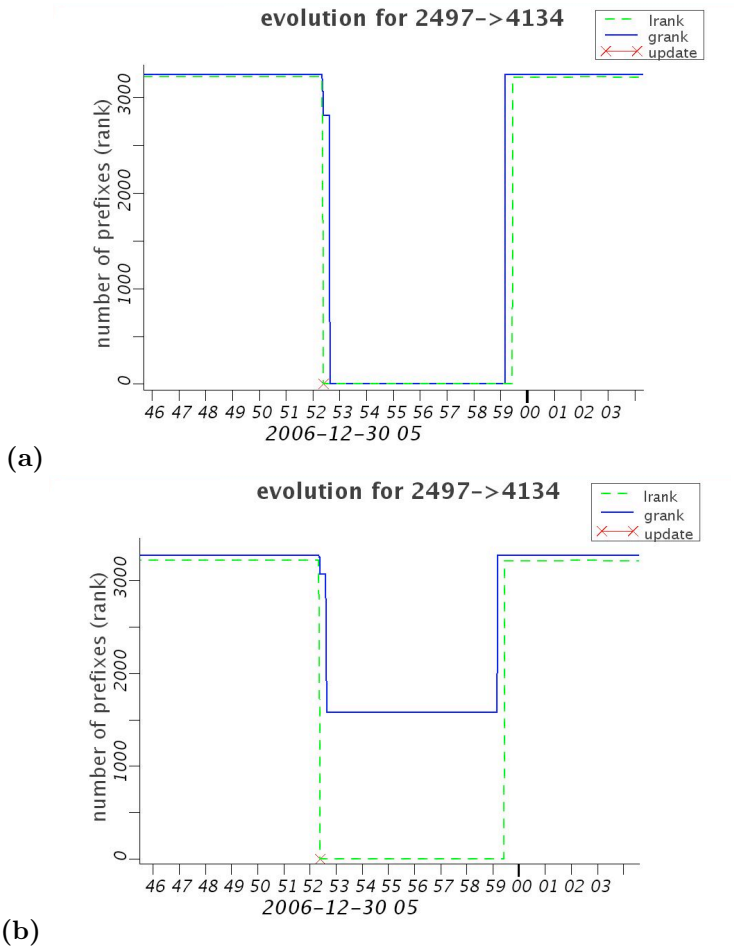


Figure 2.2: Functions *grank* (solid black) and *lrank* (dashed gray) of $(2497, 4134)$. (a) With Reliability Screening. (b) Without Reliability Screening.

$\langle 7, 8.2\% \rangle$. Hence, there are many (namely, 7) collector peers that can see e, with a similar number of prefixes. So we consider its global rank trustworthy. This analysis suggests with reasonable confidence that the path change c has been triggered by a macro-event on edge e.

This example also shows the importance of the Reliability Screening. In fact, performing the same analysis skipping such a step, we obtain the evolution of $grank(e)$ shown in Fig. 2.2.b. In this case, because of the noise generated by the unreliable RRCs, $grank(e)$ never decreases below $1,580$, making the macro-event less visible.

Theorem 1 suggests that whenever there are multiple edges with $grank = 0$ in either $c.\pi_{old}$ or $c.\pi_{new}$ the most likely responsible for the macro event is the edge closest to $c.cp$.

If the global rank analysis ends up with no candidates, we analyze each selected collector peer separately, by looking at the evolution of the local rank in $T_{c.t}$. On edges in $c.\pi_{old}$ and $c.\pi_{new}$, we search for the same patterns as above.

Generally, we trust $grank(as_1, as_2)$ more than $lrank(cp, (as_1, as_2))$, unless cp belongs to as_1 and provides its full routing table. In fact, we consider a collector peer an authoritative source of information on the AS it belongs to. Otherwise, any inference supported only by local rank analysis requires further investigation.

Section 2.6.2 describes how we compute global and local ranks.

As an example, we analyze the path change c, where $c.p = 80.124.192.0/19$, $\pi_{old} = (7575,\ 15557,\ 8228)$, $\pi_{new} = (7575,\ 2914,\ 3356,\ 15557,\ 8228)$, $c.cp = 198.32.176.177$, and $c.t =$ UTC 01/01/07 00:04:53. According to the Collector Peer Check, all the collector peers are available in $T_{c.t}$. We evaluate the global rank of all edges belonging to $c.\pi_{old}$ and $c.\pi_{new}$, and we have that $grank(e) = 0$ in $T_{c.t}$, where $e = (7575, 15557)$. Note that $\overline{grank}(e) = 148$. Unlike the previous example, the rank diversity of e is low ($\langle 2, 0.1\% \rangle$), as the edge is seen by only two collector peers, both belonging to 7575. So its global rank is not worthy. As a consequence, we analyze $lrank$ for $c.cp$. Being $c.cp$ in the left node of e, it is in the best position to observe routing events affecting e. Fig. 2.3 illustrates the evolution of $lrank(c.cp, e)$, and $grank(e)$ for $e = (7575, 15557)$ and $e' = (3356, 15557)$. It is interesting to notice that a relevant number of prefixes moves from an edge to the other (Theorem 2). From the information extracted, we can deduce with reasonable confidence that the path change c has been triggered by some macro-event affecting edge e.

2.4.3 Fine-Grained Analysis

If the Macro-Event Analysis doesn't identify any cause for the route change c, we examine flow changes in order to capture routing events which don't affect the interdomain topology. Namely, we look for events (e.g., BGP policy changes) that in general do not impact all the prefixes passing through an

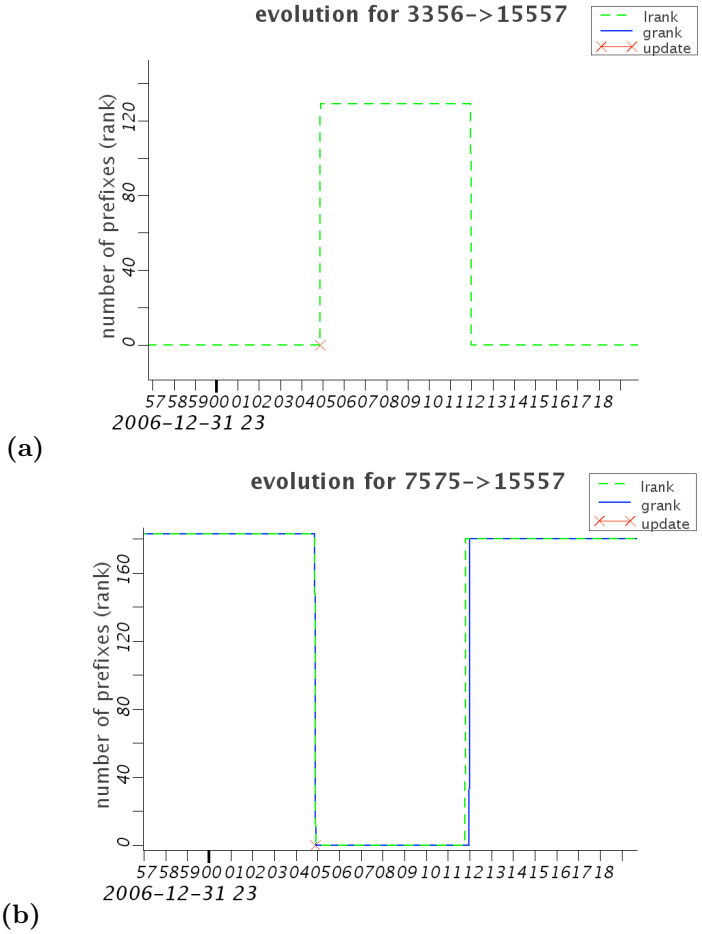


Figure 2.3: Functions *grank* (solid black) and *lrank* (dashed gray) of $(3356, 15557)$ (a), and $(7575, 15557)$ (b).

interdomain link, but only a subset of them.

In the Fine-Grained Analysis we investigate flow changes on the whole Internet. However, in our experience, a flow change can spread over a very large portion of the Internet, making the analysis unfeasible. Thus, we focus on a fraction of a flow change, introducing the concepts of path compatibility

and restricted flow.

Two paths π_1 and π_2 are *compatible* ($\pi_1 \bowtie \pi_2$) when they share a common left subsequence of at least two ASes (i.e. they share the first edge). A *restricted flow* $\Delta_t^{t+\tau}\hat{f}_P(cp,(u,w))$ is a flow defined on a subset $P \subseteq \mathcal{P}$ of the prefixes. We consider especially interesting the restricted flow on prefixes that experienced, in $T_{c.t} = [t, t+\tau]$, a change whose either the old or the new path is *compatible* with a given path π. In fact, such a restricted flow can be used to study routes coming from (moving to) π. Formally, we evaluate $\Delta_t^{t+\tau}\hat{f}_P(cp,(u,w))$, with $P = \{p \mid \ell_{cp}(p,t).\pi \neq \ell_{cp}(p,t+\tau).\pi \wedge (\ell_{cp}(p,t+\tau).\pi \bowtie \pi \vee \ell_{cp}(p,t).\pi \bowtie \pi)\}$.

For example, we analyze the path change c, where $c.p$ = 202.59.174.0/24, $c.\pi_{old}$ = (16215, 3549, 5511, 4761, 17727), $c.\pi_{new}$ = (16215, 3549, 3320, 4761, 17727), $c.cp$ = 80.81.192.143, and $c.t$ = UTC 12/27/06 10:06:17. After an unsuccessful Macro-Events Detection, we proceed with the present step.

We try to track the rearrangement of the prefixes routed away from $c.\pi_{old}$ (onto $c.\pi_{new}$). Thus, we compute the previously defined flow $\Delta_t^{t+\tau}\hat{f}_P$ where $\pi = c.\pi_{old}$ ($c.\pi_{new}$). In our example we have that prefix 202.57.0.0/24 has, in $T_{c.t}$, a path change from (16215, 3549, 5511, 4761, 17658) to (16215, 3549, 7473, 4761, 17658). The old path is compatible with $c.\pi_{old}$ ($(16215, 3549, 5511, 4761, 17658) \bowtie c.\pi_{old}$). Hence, it is part of the set P (also containing 740 other prefixes) that we use to compute the restricted flow. Observe that, in any restricted flow, edges with a positive flow value describe where prefixes leaving paths compatible with π are re-routed to. Thus, we focus on these edges to analyze prefixes that left $c.\pi_{old}$ ($\pi = c.\pi_{old}$). On the other hand, negative flow values indicate where prefixes moving on paths compatible with π come from. Therefore, we focus on these ones to study prefixes that move onto $c.\pi_{new}$ ($\pi = c.\pi_{new}$).

We build a *restricted flow graph* consisting of edges having $\Delta_t^{t+\tau}\hat{f}_P > 0$ ($\Delta_t^{t+\tau}\hat{f}_P < 0$). Fig. 2.4 outlines a sketch of the restricted flow graph computed on $c.\pi_{old}$ from our example. The graph visualizes how prefixes in P moved from edges in $c.\pi_{old}$ (red-colored, within the box) to the other edges (green-colored, outside the box). For the sake of clarity, Fig. 2.4 omits edges with negligible flow values. Notice that most prefixes move away from the first two edges of $c.\pi_{old}$. This is mainly a consequence of flow systems behavior: the flow is more likely to be high on edges closer to the source (collector peer).

Observe that a lot of prefixes move from $(3549, 5511)$ to edges $(3549, 3320)$, $(3549, 3491)$, and $(3549, 7473)$. Also, those prefixes are still routed through AS 4761 (edges $(3320, 4761)$, $(7473, 4761)$, and $(3491, 4761)$). We argue that this happens due to some event on $(3549, 5511, 4761)$, since multiple events on $(3549, 3320, 4761)$, $(3549, 3491, 4761)$, and $(3549, 7473, 4761)$ are much less

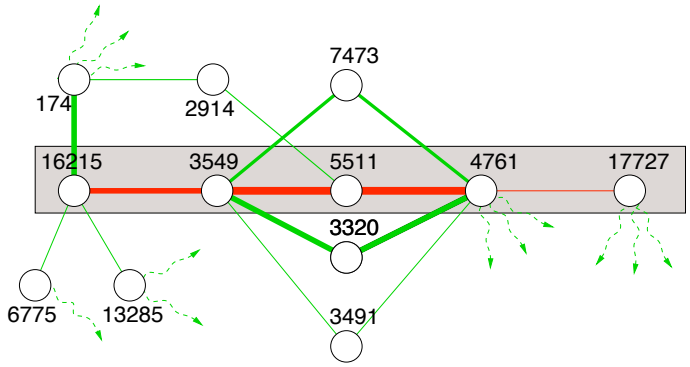


Figure 2.4: Green edges and their end-vertices are a portion of the restricted flow graph. Path $\pi = c.\pi_{old}$ is also displayed (highlighted in the box) for convenience. Thicker lines represent edges with higher value of $\Delta_t^{t+\tau}\hat{f}_P$

likely to occur concurrently. However, we cannot further distinguish if c happens because of a routing event on either $(3549, 5511)$ or $(5511, 4761)$. We generalize the above discussion by considering the nodes m_o and m_i in π having, respectively, maximum outgoing flow and maximum incoming flow in the restricted flow graph. The output set of candidates is the subpath $(m_o, \ldots, m_i)$ of π.

There are some border-line cases to consider. For example, in case two vertices have maximum outgoing (incoming) flow, we can break the tie considering the largest possible candidate set. As another example, there can be many vertices that have similar values of outgoing (incoming) flow. In this case our approach allows to deepen the analysis picking one of the path changes that involve maximum flow vertices and applying the same methodology iteratively on that change. This shift of focus makes our methodology inherently iterative, and allows to cope with the "induced instabilities" problem (see [FMM+04]), overcoming a common limitation of inference systems, which are usually able to locate causes of a route change only on the new or the old path.

2.5 BGPath: Online Tool to Support the Analysis of Network Events

In order to automatically compute the metrics our methodology relies on, we developed a BGPath, a publicly available system that combines data collected by multiple distributed monitors, checks the reliability of available data sources, and estimates the usage of interdomain links. BGPath also graphically displays detailed and aggregated information about a user-specified route change. This section describes the architecture of the service and how it supports the analysis of a route change.

2.5.1 Analyze a Route Change Using BGPath

Through the following scenario, we show how BGPath effectively supports the analysis of a route change in order to (at least partially) identify its root cause. For a complete description of the approach, see Section 2.4.

Assume that a network operator is interested in monitoring the prefix $p = 159.14.0.0/16$. In particular, he knows that, on t =Jan 1st 2007, p underwent the route change c displayed in Figure 2.5, where the old path $c.\pi_{old} = (15837, 8881, 2914, 10910, 7328)$ and the new path $c.\pi_{new}$ =(15837, 8881, 3356, 12178, 7328). Note that some exiting tools (e.g., BGPlay [bgpb, bgpc]) can be used to graphically browse through and select specific route changes.

In order to provide a network operator with a familiar representation of the portion of the Internet topology involved in the change c, BGPath draws $c.\pi_{old}$ and $c.\pi_{new}$ according to the customer-provider hierarchy (see e.g., [Gao01]). Namely, tier-1 ISPs are represented by the top nodes in the graph, and customers are placed just below according to their position in the hierarchy. In Figure 2.5, for example, the two top ASes are tier-1 providers (NTT and Level3). Besides being natural for an operator, this representation is also helpful to understand the relevance of network events related to the route change. In fact, [ZZM$^+$05] shows that events located in different levels of the hierarchy usually have significantly different impact on the network. Section 2.6.3 describes the visualization algorithm.

According to our approach, we first check the reliability of the collector peers which observed the route change c at the time t ($CP_{c,t}$). Namely, we are interested in whether the CPs $\in CP_{c,t}$ were involved, at any time close to t, in events (e.g., a reboot) that could affect the quality of the BGP data collected. The reliability information is displayed for convenience in the left panel of BGPath's main window, so the user can easily check it at a glance.

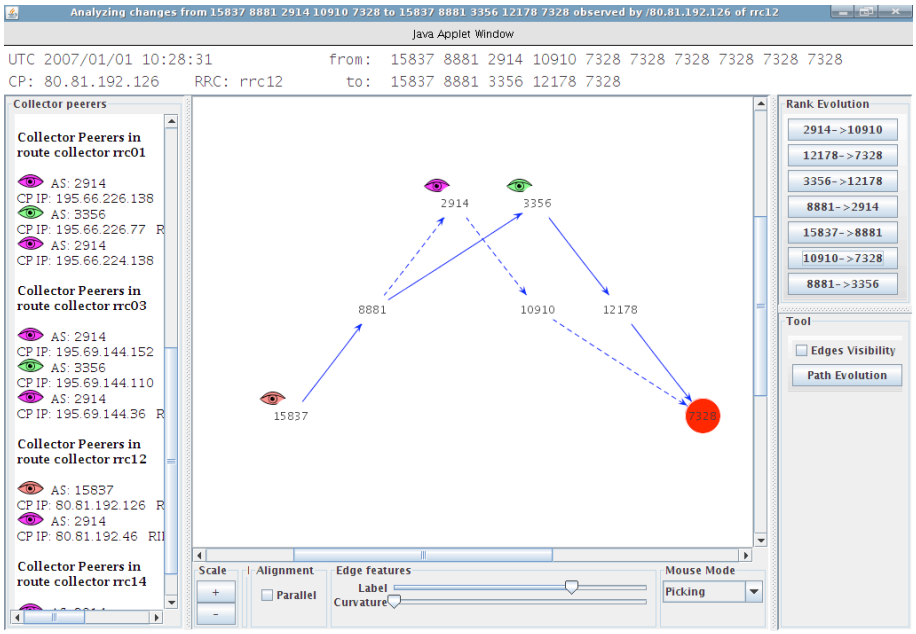


Figure 2.5: A route change c displayed by BGPath. Edges in the old path are represented by dashed lines, while edges in the new path are drawn with solid lines. Data about c are reported in the top of the window. The left panel contains a list of collector peers which observed c and the ASes they belong to, grouped by the route collectors they peer with. These collector peers are also marked by eye-shaped icons.

The ability to spot potentially unreliable data sources gives the user a high level of trust on the analysis of the change c. Thus, we believe it is a key step of our approach and a very important feature of BGPath. As far as we know, no other tools provide such an information. Note that BGPath only identifies possibly unreliable data sources, but it does not filter out the data they provide. Section 2.3 explains how we check the reliability of a collector peer by evaluating the consistency of the data it provides, while Section 2.6.1 details how to identify BGP table transfers associated with BGP session resets.

To assess the scope of the route change c, the network operator can visualize the history of all routes chosen by $c.cp$ to reach p within a fixed time window around t. Figure 2.6(a) shows the transition from $c.\pi_{old}$ =(15837, 8881, 2914,

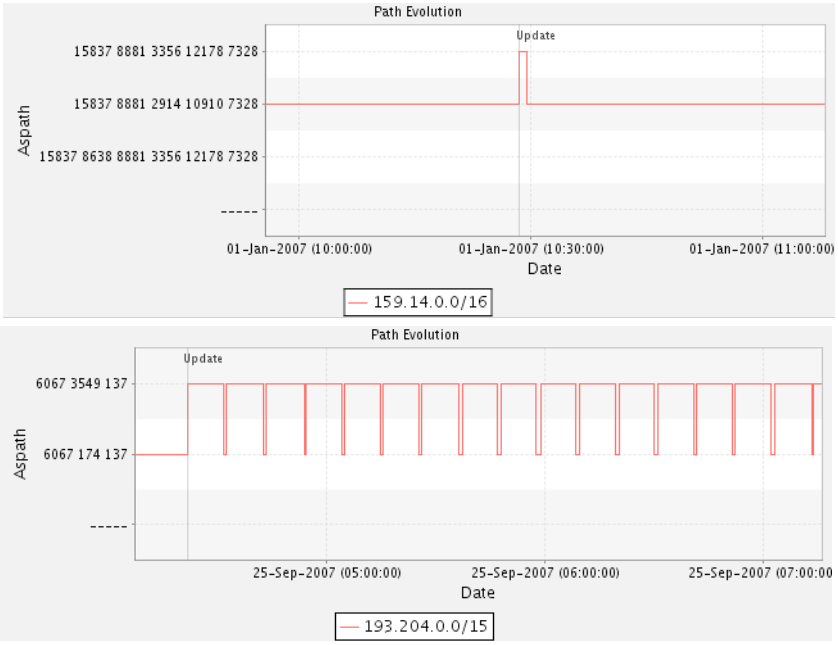


Figure 2.6: Two examples of the path evolution plotted by BGPath. Time on the x axis, a set of distinct paths on the y axis. The label "---" denotes the empty path. **(a)** A prefix experiencing a short-lived path change. **(b)** A prefix whose path changed many times within a short time interval.

10910, 7328) to $c.\pi_{new}(15837, 8881, 3356, 12178, 7328)$ at time t. Note that $c.cp$ switches path to the prefix p only twice (from $c.\pi_{old}$ to $c.\pi_{new}$ and back) within a short time interval. Instead, Figure 2.6(b) shows an unstable prefix which keeps flapping between path $(6067, 3549, 137)$ and $(6067, 174, 137)$. Hence, looking at the path evolution, it is possible to verify whether the analyzed route change is part of any specific pattern of changes, e.g. if it is just a temporary oscillation, or if it belongs to a persistent dynamics. The visualization of path evolutions is also extremely useful to detect sequences of routing changes due to BGP path exploration. This way the user can verify whether the route change c in input was induced by path exploration. Finally, the interface of the tool also allows the operator to select other prefixes announced by the same origin AS, and to plot their path evolution within the same chart. Thus an ISP can monitor the

behaviors of different prefixes it announces on the network, and the impact of per-prefix routing policies (e.g., interdomain traffic engineering configurations).

Once assessed the reliability of data sources and the scope of the route change c, the network operator can analyze the evolution of the number of prefixes routed through every edge e affected by c (i.e. belonging to either $c.\pi_{old}$ or $c.\pi_{new}$) in order to have a rough estimate of the traffic load born by e. BGPath plots both *grank* and *lrank* evolutions for all visualized edges. For example, Figure 2.7(a) shows that, according to both *lrank* and *grank*, $e = (10910, 7328)$ carries a steady quantity of prefixes over time, but exhibits a discontinuity right at time $c.t$. For a couple of minutes, in fact, all the prefixes that passed through e moved somewhere else. Since more than 20 collector peers contribute to the global rank, this discontinuity is an evidence of some problem affecting e. Zooming in the $lrank(e)$ plot, Figure 2.7(b) exhibits lack of synchronization between *lrank* and *grank*. Note that, in general, *grank* experiences a delay in recording a "negative" variation, while it is much more reactive in recording a "positive" variation, with respect to *lrank*. This is a consequence of the aggregated nature of *grank*: since it is defined to be a set-theoretical union, $grank(e)$ does not drop to a lower value until all collector peers stop using e to reach at least one prefix. Since worldwide spread collector peers are not perfectly synchronized, the *grank* is bounded to the slowest-reacting collector peer. Symmetric considerations apply to "positive" variations. Figure 2.7(c) exemplifies *grank* and *lrank* of a link probably experiencing some faults.

At this point, the network operator has an evidence of some event happened on e. This event caused, for a very short time interval, all the prefixes passing through e to change their routes. Prefix $c.p$ is among them. The user can further deepen his analysis by exploring different time intervals, in order to check if the shortage occurred again.

2.5.2 A Stream-Based Approach to Process Inter-domain Data

The approach described in Section 2.5.1 relies on the availability of information such as local and global ranks of all the interdomain links in Internet and reliability of all the available collector peers. This information is not explicit in BGP raw data (i.e., RIB dumps, updates, and session logs), and efficiently computing it is critical for a near real-time analysis of route changes. This section provides a high level description of how BGPath processes the input data within reasonably strict time constraints. Section 2.6 details the algorithms we

2.5. BGPath: Online Tool to Support the Analysis of Network Events

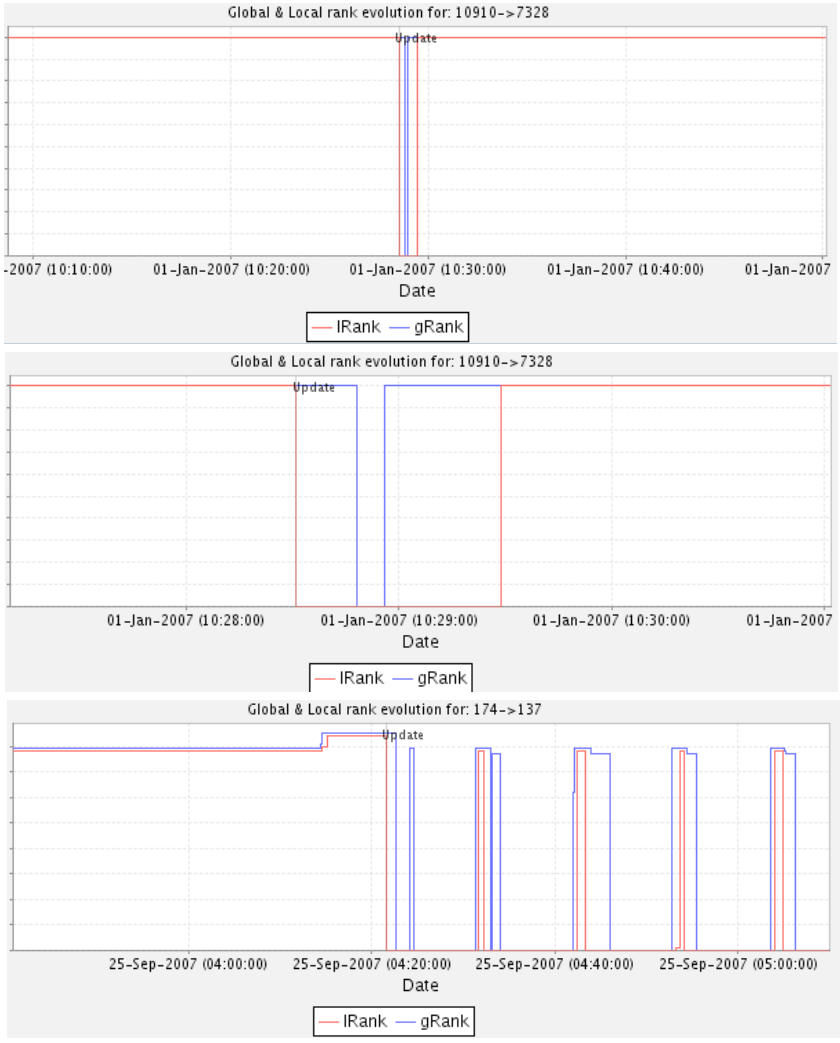


Figure 2.7: Local and Global ranks. **(a)** A link with a short service discontinuity. **(b)** Zoom of the discontinuity. **(c)** A link with some malfunction.

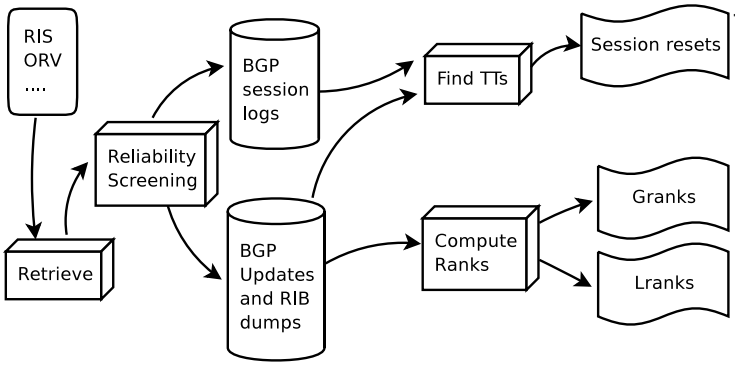


Figure 2.8: Information flow of the computation process.

devised and their performance evaluation.

The key idea to speed up the computation is treating input and output data as streams: this way, the current value of the metrics is incrementally updated and pushed into the output stream at a little cost, in terms of both time and memory requirements. As a drawback, this approach requires to scan the output stream in order to access a specific part (e.g., when searching for the *grank* of an edge at a specific time), potentially decreasing response time. We deal with this issue by partitioning and indexing the output stream.

Figure 2.8 outlines the building blocks of the computation process. Namely, the input data stream is handled as follows: (*i*) *Retrieving*: RIB dumps, updates, and session logs (if available) are collected from several data sources. (*ii*) *Reliability Screening*: collector peers are periodically checked for consistency. Those found inconsistent are temporarily disregarded (see Section 2.3). (*iii*) *Computing Ranks*: taking consistent BGP routing data (RIB dumps and updates) as input, the algorithms described in Section 2.6.2 compute local and global ranks for all the collector peers and for all the edges in the Internet. (*iv*) *Finding Table Transfers*: in parallel, table transfers and their alleged duration are identified by applying the algorithm in Section 2.6.1 to consistent BGP data. Then, identified table transfers are combined with session logs (if available).

Note that, since the available data sources currently provide BGP updates grouped into chunks, BGPath has to fully process a chunk before another one

comes out of the input stream, in order to satisfy the strictest time constraint. Given the average per-chunk performance reported in Section 2.6, BGPath is able to process a 15-minute chunk in less than 3 minutes, including the time to retrieve the input data.

Also, the space to store both raw and computed data is an important constraint. A week of data costs approximately 65GB. Observe that data compression would sacrifice CPU time for space, delaying data processing and slowing down query responses. To limit storage requirements, BGPath currently manages data within a fixed-length sliding window spanning back up to n days. Currently, we keep one month of data (i.e., $n = 30$) available for user queries.

We evaluated the performance of our tool using a Linux testbed with the following hardware: 2x Intel Xeon 2.80GHz CPUs, 512KB cache, and 4GB RAM. Note that this is an average platform, thus any common machine can effectively run BGPath.

2.6 Algorithms

This section describes the algorithms we devised to identify table transfers and compute local and global ranks, that fulfill the time constraints outlined in Section 2.5.2. Performance analyses illustrated in this section will show that both the algorithms have reasonable time and memory requirements to be executed on an average machine. We also describe how to visualize AS-path changes in order to support the analysis of network events.

2.6.1 Identification of Table Transfers

[WZP+02] shows that BGP sessions between route collectors and collector peers can undergo frequent resets. Observe that, after the fault of a session between a collector peer *cp* and its route collector *rc*, the data collected by *rc* are out of sync with *cp*, until the session is restored and *cp* sends its whole BGP table to *rc*. Hence, rank values can be misleading in the time interval between the fault and the end of the subsequent table transfer. Note that this issue especially affects local ranks, while global ranks are less sensitive to the contribute of a single collector peer.

When available, BGP state messages provide an evidence of session resets by recording transitions from/to the BGP state `6=Established` [RLH06]. Unfortunately, only RIS collector peers supply state messages. Thus, we devised Algorithm 1 in order to identify, with a reasonable accuracy, all the BGP table

transfers occurred between collector peers and route collectors in a given set of route changes. We stress that our algorithm is valuable even when state messages are available, because it also estimates the duration of a table transfer. This information cannot be extracted from state messages and it is necessary for pinpointing updates caused by resets to disregard them during the analysis.

Although [ZKL+05] has already described an approach to identify table transfers, we found out that it does not scale over a large set of collector peers. Namely, even if [ZKL+05] does not formalize the computational complexity analysis of the proposed algorithm, it is easy to find that it requires $O(n\omega\sigma)$ time and $\Omega(\omega\sigma)$ space, where n is the number of processed changes, σ is the maximum number of changes per second, and ω is the width of a time window that is used to scan the changes (whose maximum size is two hours). Although time complexity is feasible if $\omega\sigma$ is $o(n)$, processing a huge set of data requires a lot of memory. Algorithm 1 was designed to tackle this space complexity problem.

Given a stream of route changes, Algorithm 1 pinpoints a table transfer from any collector peer, with an approximation of the start time of the transfer and its duration.

The main intuition is that, within a reasonably short time interval, a collector peer usually sends updates for a set of prefixes that is relatively small compared to its full BGP table. When most of the full RIB is announced within a short time, we guess the occurrence of a table transfer. Algorithm 1 slides a fixed-width time window over the BGP update stream and compares the number of distinct prefixes sent by a collector peer in the window to its full RIB size.

Since the RIB size varies over time, we define a function $\rho(cp, t), \rho : (\mathcal{CP} \times \mathcal{T}) \rightarrow \mathbb{R}$ that accounts for the evolution of the RIB size of cp, and a threshold ratio δ. Every time that the window contains more than $\delta\rho(cp, t)$ distinct prefixes, the algorithm alerts an alleged table transfer. We define ρ as the weighted average of the RIB size of cp until time t, where the weights are the amounts of time within which the RIB had a certain size. This choice has at least two advantages. First, it is computable in $O(1)$ time, without memory penalties (only one value per collector peer needs to be retained in memory). Second, it is low-sensitive to short-lived hijacking, even when they involve many prefixes. Namely, let t_i be the time of the i-th change of the RIB size of cp, let $R(t_i)$ be the size of that RIB at time t_i, and let t_0 be the first considered time,

then

$$\rho(cp,t) = \frac{\sum_{i=1}^{k} R(t_{i-1}) \cdot (t_i - t_{i-1})}{t - t_0}.$$

We distinguish between *pumping* and *vacuum* table transfers. The first is a large set of announcements from a router that sends a neighbor its own full routing table. The latter is an explicit withdrawal of a large set of prefixes.

Algorithm 1 illustrates all the steps with a pseudo-code notation. For each collector peer, we keep in memory a set of known prefixes (RIB_{tt}), and two dictionaries, P_W and P_{RA}, that map a prefix to the latest time it was withdrawn and re-announced[1], respectively. Whenever an change c is received, we classify it as a new announcement, a withdrawal, or a re-announcement. In the first case, the prefix is inserted into the current RIB_{tt}, as shown in line 2. In the other cases, the time value associated with $c.p$ in P_W or in P_{RA} is updated with $c.t$ (lines 6 and 15). Lines 7 to 9 (16 to 18) remove from P_W (P_{RA}) prefixes that were withdrawn (re-announced) more than ω_w (ω_r) seconds before $c.t$. Finally, in line 10 (19) we compare P_W (P_{RA}) to $\delta\rho(c.cp, c.t)$, in order to evaluate whether the number of prefixes in P_W (P_{RA}) approximates the average size of RIB_{tt}. If so, we detect a vacuum (pumping) table transfer.

In our experiments we set $\delta = 0.99$ according to the results in [ZKL+05], and we chose $\omega_r = \omega_w = 500$ seconds.

Using Algorithm 1, we are able to identify occurring table transfers in a single sweep of all the changes ($O(np)$ time, where n is the number input route changes and p is the number of all known prefixes), only using $O(p)$ memory space (considering constant the number of collector peers). In the following we show the analysis of the algorithm complexity. First, we assume $O(1)$ time for all the operations on hash tables (e.g. insertion and deletion of elements in a set, retrieval of the value associated to a specific key, evaluation of the cardinality of sets). As shown before, evaluating $\rho(c.cp, c.t)$ takes $O(1)$ time. Without loss of generality, we only describe the complexity for processing input withdrawals. Similar arguments hold for re-announcements. Note that $\min_p P_W(p)$ is never actually computed, as its running value can be kept in a proper variable min_W. Hence, the whole complexity is bounded by the step that drops outdated entries from the map P_W (lines 7 to 9). In the worst case and without further performance optimization, we scan all prefixes in

[1] In this context, a re-announcement is a change c such that both $c.\pi_{new}$ and $c.\pi_{old}$ are non-empty

loa 1: Identify Table Transfers of a Collector Peer

Require: a route change c is recorded from $c.cp$ at $c.t$
1: **if** $c.\pi_{old} = \phi$ **then** // c is new announcement
2: $RIB_{tt} \leftarrow RIB_{tt} \cup c.p$
3: **else** // c is a withdrawal or a re-announcement
4: **if** $c.\pi_{new} = \phi$ **then** // c is a withdrawal
5: $RIB_{tt} \leftarrow RIB_{tt} - c.p$
6: $P_W(c.p) \leftarrow c.t$
7: **for all** p such that $P_W(p) < c.t - \omega_w$ **do**
8: remove p from P_W
9: **end for**
10: **if** $|\text{keys}(P_W)| > \delta\rho(c.cp, c.t)$ **then**
11: OUTPUT a *vacuum* Table Transfer occurred within $[\min_p P_W(p), c.t]$
12: clear P_W
13: **end if**
14: **else** // c is a re-announcement
15: $P_{RA}(c.p) \leftarrow c.t$
16: **for all** p such that $P_{RA}(p) < c.t - \omega_r$ **do**
17: remove p from P_{RA}
18: **end for**
19: **if** $|\text{keys}(P_{RA})| > \delta\rho(c.cp, c.t)$ **then**
20: OUTPUT a *pumping* Table Transfer occurred within $[\min_p P_{RA}(p), c.t]$
21: clear P_{RA}
22: **end if**
23: **end if**
24: **end if**

keys(P_W). Observe that keys(P_W) never contains more than p distinct prefixes. Thus, the worst case time complexity is $O(np)$. Moreover, using min_W, the step in lines 7 to 9 is only necessary when $min_W < u.t - \omega_w$. Space complexity is trivially $O(p)$ since we keep in memory RIB_{tt}, P_W, and P_{RA}. Thanks to the improvement in space requirements, we can account for table transfers of all collector peers at once.

To compare the effectiveness of Algorithm 1 against [ZKL+05], we fed both algorithms with the same week of BGP data from a RIS collector (namely, rrc00) peering with 13 full CPs and we ran them on a single-processor ma-

chine. Since [ZKL+05] is designed to process one monitor in each execution, we launched 13 executions of the algorithm in parallel, in order to exploit caching and minimize waits. Although the two algorithms identified exactly the same set of table transfers, Algorithm 1 took 345 seconds and 130MB of memory, while [ZKL+05] took 480 seconds and 750MB overall. Summarizing, Algorithm 1 saved 28% time and 83% memory with respect to [ZKL+05]. On the other hand, the algorithm in [ZKL+05] can be more accurate in identifying the exact beginning of a table transfer, because it performs several backward scans. Table 2.1 illustrates the performance of Algorithm 1, ran over the reference week.

Execution (wall clock) time	11:30:32 (hh:mm:ss)
Memory used (peak)	1.2 Gbyte
15' chunk avg processing time	62 sec

Table 2.1: Time and memory performance

2.6.2 Computation of Local & Global Ranks

Looking at the evolution of local and global ranks over time allows network operators to monitor the usage of interdomain links and peerings, and helps pinpoint macro-events which affect either the physical or the logical network topology (e.g. interdomain link faults/restorations, BGP router faults/restorations, or BGP peering shutdowns/setups). Such events can have a dramatic impact on Internet routing. Section 2.4.2 describes how to infer interdomain macro-events by analyzing *lrank* and *grank*.

We define Algorithm 2 to compute the global rank of all edges. Local ranks can be computed with simple reference counting, so we omit the description of the algorithm. We keep a running counter $\lambda(edge, c.p)$ that indicates of the number of distinct collector peers whose current route to prefix $c.p$ contains *edge*. Whenever an change c is received, the counters are decreased for edges in $c.\pi_{old}$, as shown in line 3, while they are increased for edges in $c.\pi_{new}$ (line 12). The global rank of *edge*, $grank(edge)$, is in turn decreased (line 5) whenever its λ reaches 0, i.e. whenever 0 collector peers are using *edge* to reach $c.p$. Similarly, whenever *edge* is used again to reach $c.p$ by at least one collector peer, $grank(edge)$ is increased at line 14.

Algorithm 2 computes *grank* in $O(nl)$ time and $O(ep)$ space, where n is the number of input route changes, l is the maximum length of an AS-path, e is the

loa 2: Compute Global Ranks

Require: a route change c is recorded from $c.cp$ at $c.t$
1: **if** $c.\pi_{old} \neq \phi$ **then**
2: **for all** $edge \in c.\pi_{old}$ **do**
3: $\lambda(edge, c.p) \leftarrow \lambda(edge, c.p) - 1$
// $c.cp$ does not use $edge$ any more to reach $c.p$
4: **if** $\lambda(edge, c.p) = 0$ **then**
5: $grank(edge) \leftarrow grank(edge) - 1$
// no cp uses $edge$ to reach $c.p$
6: **end if**
7: OUTPUT $grank(edge)$, $c.t$
8: **end for**
9: **end if**
10: **if** $c.\pi_{new} \neq \phi$ **then**
11: **for all** $edge \in c.\pi_{new}$ **do**
12: $\lambda(edge, c.p) \leftarrow \lambda(edge, c.p) + 1$
// $c.cp$ now uses $edge$ to reach $c.p$
13: **if** $\lambda(edge, c.p) = 1$ **then**
14: $grank(edge) \leftarrow grank(edge) + 1$
// a new cp starts using $edge$ to reach $c.p$
15: **end if**
16: OUTPUT $grank(edge)$, $c.t$
17: **end for**
18: **end if**

number of edges in Internet, and p is the number of all known prefixes. In fact, implementing λ and $grank$ values with hash tables and supposing $O(1)$ time for hash table operations, the time complexity of Algorithm 2 only depends on the number of input changes and on the number of edges contained in the paths $c.\pi_{old}$ and $c.\pi_{new}$. Observe that no change is considered twice, i.e. the algorithm performs no backward scans when elaborating a stream of updates. Thus, it requires $O(nl)$ time. Note that, because of the high connectivity of the Internet, we consider l as a constant value. About space complexity, Algorithm 2 keeps in memory a counter λ for each distinct edge and prefix ($O(ep)$), and the running $grank$ value ($O(e)$). Note that similar arguments apply to the algorithm that computes local rank. In that case, the space complexity depends on the number of existing collector peers. Anyway, since

the addition of a new collector peer is relatively rare, the number of collector peers can be safely considered as a constant.

[LMZ04] provides a publicly available tool for visualizing local ranks, but it uses data only from a limited set of collector peers. Unfortunately, since [LMZ04] does not describe in detail how local ranks are computed on the server-side, we can not make a performance comparison with their approach.

For the sake of efficiency, we implemented the algorithms for computing *lrank* and *grank* within a single program, using shared data structures. Our experience shows that this optimization can efficiently handle the whole input data stream. Also, note that the elaboration can run in parallel on multiple machines by simply partitioning the prefix space, assigning a subset of prefixes to each machine, and, finally, summing up the partial *lrank*s and *grank*s computed by each machine. This setting clearly improves the scalability of our approach. Table 2.2 reports the performance of the algorithms, running the single-program implementation over the reference week.

Execution (wall clock) time	13:48:43 (hh:mm:ss)
Memory used (peak)	2.7 Gbyte
15' chunk avg processing time	74 sec

Table 2.2: Time and memory performance

2.6.3 Visualization of AS-path Changes

As already stressed above, the Internet is renowned to be highly dynamic as AS-paths to any destination may frequently change (from an old path to a new path). Such AS-path changes may impact the Internet operation and are usually debugged manually.

Figure 2.9(a) shows the AS-paths (valid at a specific time) from a set of ASes to a specific destination, as displayed by BGPlay [bgpb, bgpc]. After a few seconds the state of the network can be significantly different (see Figure 2.9(b)). Due to the enormous amount of AS-path changes occurring in a short time period, it is very difficult to spot them and, thus, to locate their root causes. Hence, effectively visualizing an AS-path change can significantly help understand the Internet dynamics.

We propose to focus the visualization on a single AS-path change and to display it on the customer-provider hierarchy, as [ZZM+05] shows that network events located at different levels of the hierarchy have usually significantly

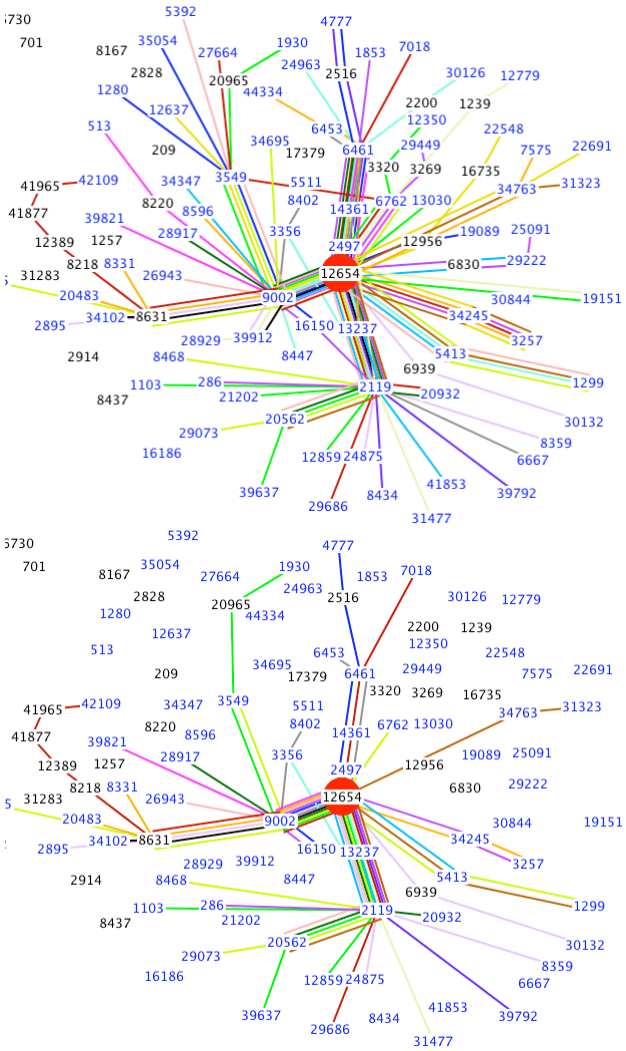


Figure 2.9: AS-paths towards a specific destination. Each number represents an AS. The red node is the destination.

different impact on the Internet. Namely, our algorithm to draw an AS-path change on the customer-provider hierarchy 2.10(0) consists of the following steps:

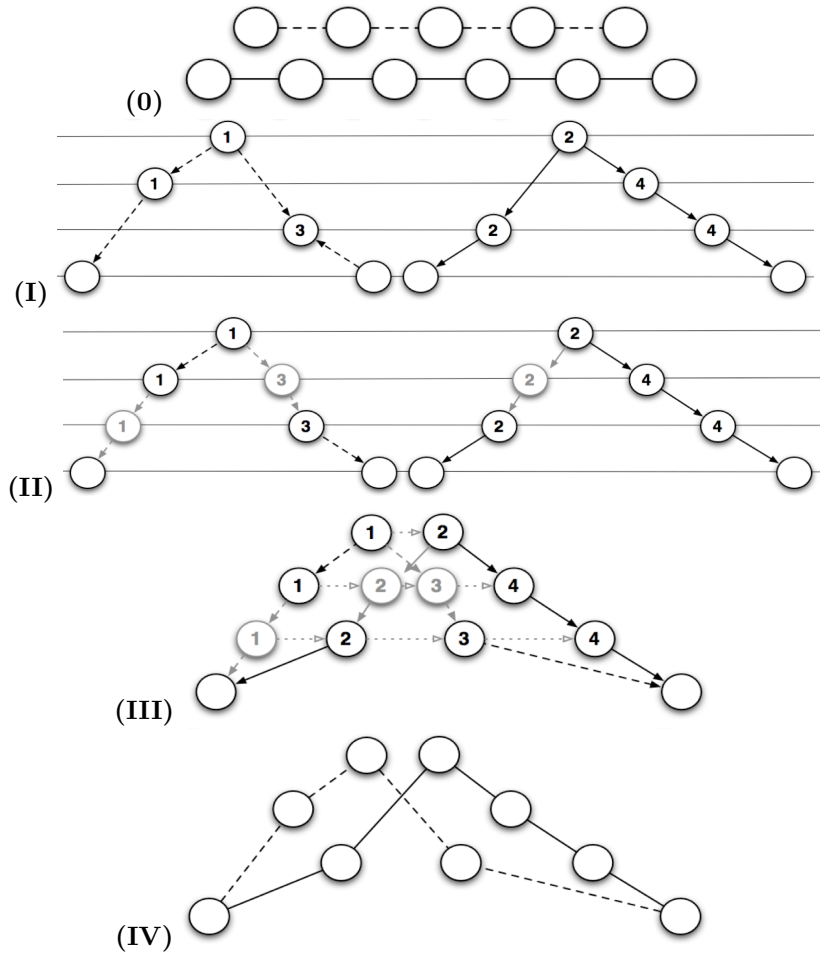


Figure 2.10: Our algorithm to visualize an AS-path change.

(*i*) First, we assign customer-provider relationships to the links of both old and new AS-paths, and we direct the links from providers to customers. As shown by [Gao01], we can then classify ASes of both paths according to the *valley-free property* as follows:

type 1 : nodes in the "uphill" portion of the old path

type 2 : nodes in the "uphill" portion of the new path

type 3 : nodes in the "downhill" portion of the old path

type 4 : nodes in the "downhill" portion of the new path

We assign vertical coordinates using a topological sort of the graph 2.10(I).

(*ii*) We then compute the horizontal coordinates. Namely, we first split edges spanning over multiple vertical layers by adding extra nodes and edges 2.10(II). Further, we add extra edges between nodes on the same layer, from nodes with lower type values to nodes with higher values 2.10(III).

(*iii*) Finally, the topological sort of this augmented graph provides us with the horizontal coordinates, such that nodes in the new path are placed right of nodes in the old path, according to the common intuition of time flowing left-to-right 2.10(IV).

Figures 2.11 show how BGPath displays sample AS-path changes.

2.6. Algorithms

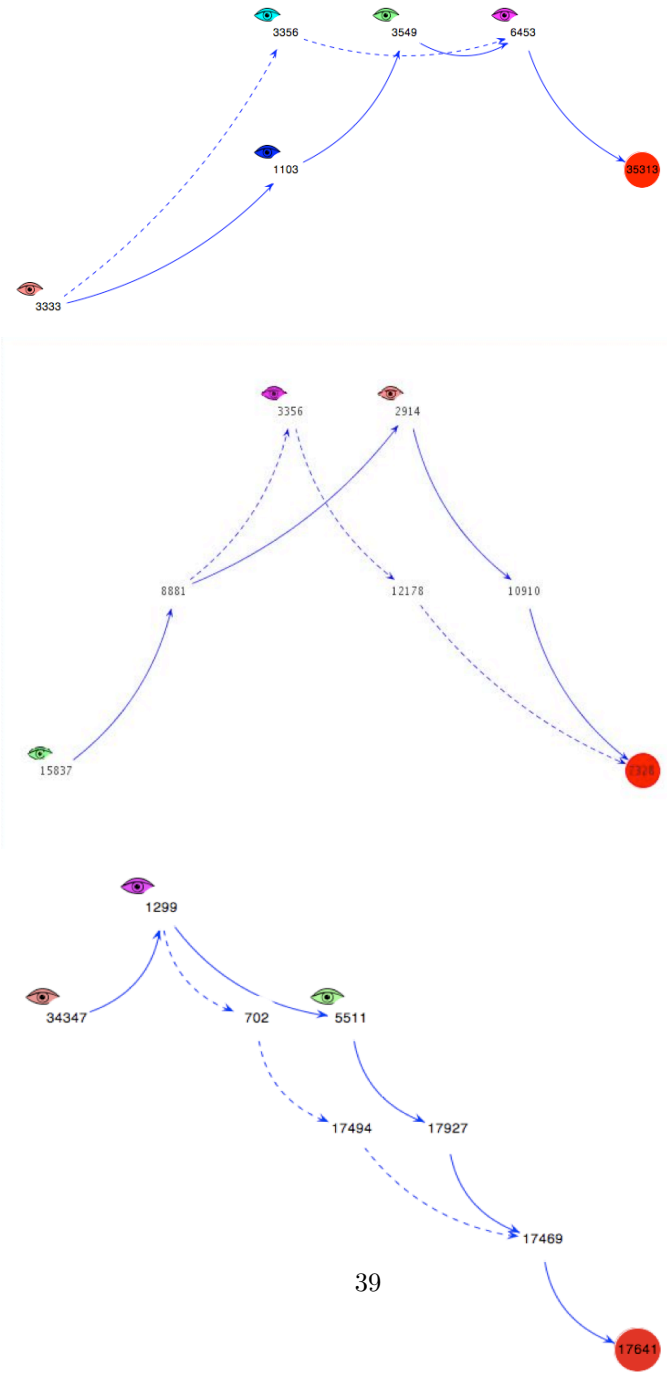


Figure 2.11: AS-path changes displayed by BGPath. Solid edges belong to the new path, dashed edges belong to the old path.

2.7 Validation

Validating the effectiveness of methodologies for root-cause analysis is a well known hard task both using Internet real data and working on simulations. In the first case, it is difficult to rely on a complete and meaningful set of faults, since producing worldwide outages is of course unfeasible and there is no publicly available history of past faults. In the second case, reconstructing a realistic scenario with the existing platforms is a challenging task since most of them approximate some dimensions (e.g., topology, policies, routing dynamics) of the problem in order to spare resources. We approached the problem from the simulation perspective, supporting it with the analysis of real Internet data to tackle simulation limitations.

2.7.1 Evaluation through Internet-scale Simulation

We performed extensive simulations, using the state-of-the-art C-BGP platform [QU05]. Namely, we settled a network with the Internet topology from [DKF+07], inferred using RV data [roua] collected on August 2007. We set up BGP policies according to the customer-provider relationships provided therein and we successfully checked the validity of the policies with the methodology in [DEH+07]. The resulting AS graph consisted of 25,599 ASes and 52,135 interdomain links. In order to account for the impact of collectors' location, we placed collector peers in the same ASes as the full collector peers of RV.

C-BGP handles efficiently only a small number of prefixes on such a huge topology, hence we were able to only deal with about 400 prefixes at the same time. Each prefix was originated by a different AS. To reduce the bias due to the location of the originating ASes, we randomly selected the set of originators in 12 distinct and independent experiments.

In each experiment, we separately generated 36 routing events and gathered all the updates collected by our collector peers. We applied our methodology to this dataset and then we compared the output candidate set with the actual root cause of each event. We simulated 3 different types of events: (i) interdomain link failures/restorations, (ii) routing policy (local-pref) changes, enforced by an hard-reset, and (iii) routing policy (local-pref) changes, enforced by soft-reset [Che00]. Routing policy changes were configured such that they only affected a subset of the prefixes. We further classified events according to their location in the Internet hierarchy (tier1/transit/stub ASes), choosing 3 distinct affected edges for each class.

2.7. Validation

Table 2.3 summarizes our results. Each entry of the table shows the accuracy of our approach for the given event type. Percentages represent the ratio between the number of input updates for which the methodology correctly returned a candidate set containing the root cause, and the total number of updates triggered by the event. The ratios were averaged over the 3 edges and the 12 distinct choices of the originators. Fig. 2.12 shows the Cumulative Distribution Function (CDF) of the size of the returned candidate sets. Most sets had 1-2 elements.

Table 2.3: Percentage of updates correctly related to an event. LP = local preference change, T1 = tier1 AS, t = transit AS, s = stub AS

Event type	**# of updates**	**Macro**	**Fine**	**Missed**
link down-up, T1-T1	10303	100%	0%	0%
link down-up, T1-t	8898	100%	0%	0%
link down-up, t-t	7358	100%	0%	0%
link down-up, t-s	12855	100%	0%	0%
LP hard, T1-T1	6998	71.28%	27.82%	0.9%
LP hard, T1-t	5882	99.9%	0%	0.1%
LP hard, t-t	4759	100%	0%	0%
LP hard, t-s	8732	99.9%	0%	0.1%
LP soft, T1-T1	8542	0%	92.58%	7.42%
LP soft, T1-t	1340	0%	98.5%	1.5%
LP soft, t-t	1704	0%	100%	0%
LP soft, t-s	1056	0%	95.17%	4.83%

Table 2.3 shows that the Macro-Events Detection step is quite effective in explaining updates generated by topology changes. The results are also quite satisfactory for the Fine-Grained Analysis. Notice that policy changes requiring hard reset have been approximately seen as topology changes and hence detected by the Macro-Events step. Results show that it is more difficult to detect events which do not alter the network topology (e.g., LP soft), because they can affect only a subset of all the prefixes on a link. As shown by [FMM+04], events involving the T1 network are usually harder to locate, due to the huge redundancy within the Internet core.

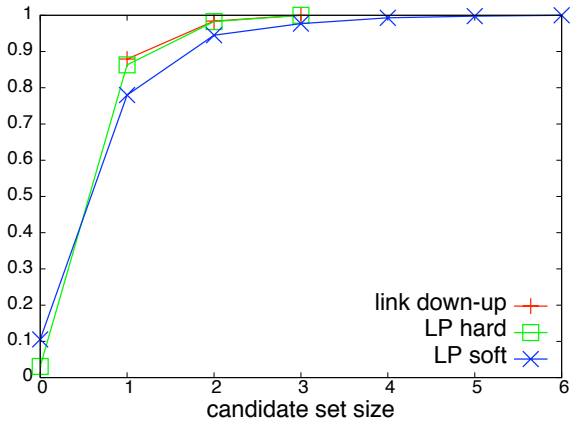


Figure 2.12: CDF of the size of the candidate sets.

2.7.2 Experimental Results

The confidence in our experimental validation is supported by [QU05], which provides some evidence that C-BGP is a good simulator of real Internet. However, the results are affected by the following limitations: (i) Collector peers were all reliable. This does not allow to understand the impact of the Reliability Screening and of the Collector Peer Check. (ii) C-BGP is optimized to reduce path-exploration updates. This decreases the relevance of the link down-up experiments. In fact, in the real world a negative event (e.g., a link-down) produces several path-exploration updates.

In order to better understand the impact of limitation (i), we analyzed real-world data of the reference week. First, we looked for all collector peers affected by reboots, performing the Collector Peer Check (Section 2.4.1). We used only route collectors that successfully passed the Reliability Screening (Section 2.3.1). Taking into account only full collector peers, we identified 90 table transfers, each one corresponding to a session reset. BGP session state messages, only available for RIS collectors, reported 71 session resets. Overall, the average percentage of time affected by session resets was 0.01% of the reference week per each collector peer. Note that the average increased to 3% if we considered both full and partial collector peers. These results show that the Collector Peer Check step discarded a non negligible portion of the input data.

2.7. Validation

Table 2.4: Number of edges affected by a grank0 event.

	reliable cps	**all cps**
all edges	8528	6332 (-25.75%)
edges with $\overline{grank}(e) > 50$	455	372 (-18.24%)
edges with $\langle n \geq 3, \sigma_x/\overline{x} \leq 30\% \rangle$	16	13 (-18.75%)
edges with $\overline{grank}(e) > 50$ **&** $\langle n \geq 3, \sigma_x/\overline{x} \leq 30\% \rangle$	9	7 (-22.22%)

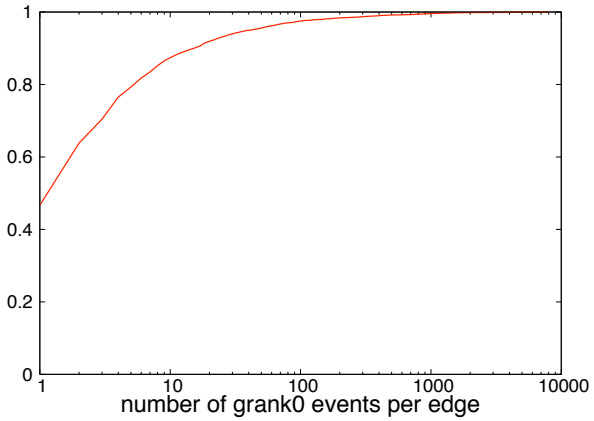


Figure 2.13: CDF of the number of grank0 events that each edge experienced.

Afterwards we applied the Macro-Events Detection step on the filtered data. Table 2.4 shows the number of distinct edges e affected by at least one *grank0 event*, i.e. with $\overline{grank}(e) > 0$ and $grank(e,t) = 0$ for at least one t in the reference week. We detected 8,528 edges, revealing that grank0 events have been significantly frequent. Among those edges, Fig. 2.13 shows the CDF of the number of grank0 events experienced by each edge. Out of the detected edges, 455 were relevant edges, i.e. edges with high grank. Only a few edges had a global visibility, i.e. had a significant rank diversity. The second column of Table 2.4 points out that some grank0 events were not detectable without eliminating unreliable collector peers.

Concerning limitation (ii), we notice that path-exploration updates are perhaps the less interesting and are quite easy to spot, since they usually can be

grouped into sequences of transient path changes that have a very short duration.

To have an idea of the impact of this step of the methodology, it is interesting to examine the results presented in [MFM+06]. That paper studies the distribution of the number of distinct AS-paths observed in the Internet between each AS pair. Such distribution puts in evidence that this kind of "path diversity" happens with high frequency.

2.7.3 Comparison with Previous Work

Comparison with previous work is not trivial since most of the approaches (e.g., [CSK03, XCZ05]) try to locate events, with little or no interest in the association between an event and the updates it triggered. The simulation in [XCZ05] does not realistically represent the Internet (400 ASes, one vantage point, and shortest path routing), so results are mostly incomparable. Ref. [FMM+04] simulates link down-up events, using a setting similar to ours, and obtains 89% accuracy, while we achieve 100% for the same event type.

2.8 Conclusions

Despite the large amount of efforts, finding the causes of specific interdomain routing changes is extremely challenging. On one hand, nowadays both researchers and network administrators can benefit from large BGP datasets, provided by several BGP collectors spread worldwide. On the other hand, existing approaches exploiting such data strive to identify all network events, disregarding specific changes.

This chapter gives three fundamental contributions to face this problem. First, we showed a new, clean model for describing BGP updates based on a flow system. Second, relying on the model, we presented a methodology that tackles the root cause analysis of a specific route change. We evaluated our approach through Internet scale simulations and real world data analysis. Third, in order to support the methodology, we developed BGPath, a publicly available system that analyzes a user-specified route change and extracts related information from huge amounts of BGP data. Specifically, BGPath combines data collected by multiple distributed monitors and checks the reliability of available data sources. Then, it estimates the usage of each interdomain link, both from local and global views, taking into account all routed prefixes. Finally, BGPath graphically displays detailed and aggregated information about

the route changes experienced by user-specified prefixes. We showed how BGPath processes in a streaming fashion a huge amount of BGP data, we detailed the underlying algorithms, and we discussed their time and space complexity. Furthermore, we walked through a real use case to illustrate how BGPath supports the analysis of a specific route change.

Chapter 3

Identifying Contributors of Routing Dynamics using Multiple Views

The Internet routing infrastructure is a large scale distributed system where routing changes occur all the time and occasional instabilities of important links can disturb routes to large numbers of destinations. Complex interconnections of the system and sophisticated routing policies lead to different vantage points observing different routing changes, making it difficult to identify major events or pin down their origins.

In this chapter, we measure the changes of routes carried over each AS and AS-link as observed from multiple monitors and apply Principal Component Analysis to identify the most involved ASes or links in the routing system. We show through simulations and case studies that our approach can effectively detect large scale routing changes and locate their major contributors.

The main results presented in this chapter are also described in [ACK+08].

This chapter is organized as follows. Section 3.1 presents previous works and summarizes our contributions. Section 3.2 explains how we use PCA over the routing data. Section 3.3 reports our simulation experiments under controlled settings to understand how accurately we can identify contributors using our technique. In Section 3.4, we apply our scheme to real BGP data over two weeks to identify the main contributors of routing dynamics. Section 3.5 concludes the chapter.

3.1 Introduction

Routing dynamics identification and diagnosis is difficult for several reasons. First, the sheer size of the system: there are about 30,000 ASes in the global routing system. Second, multiple routing events can occur simultaneously. Furthermore, due to complex interconnects among ASes and routing policies, different vantage points have different views of routing changes. This large data set is inherently multi-dimensional, making it very difficult to identify whether any major routing events occur, or where they occur.

Previous Work

In a seminal work regarding network instability, [LMJ99] identifies several causes of routing instabilities in the Internet, without however diagnosing their topological origin. Later efforts [CGH03, CSK03, FMM+04, WMRW05] analyze BGP updates by aggregating data along one of the three dimensions: time, monitors and prefixes, to obtain the candidate sets of routing instability origins. [TR04] describes a framework to detect the cause of a routing change using a coordinated diagnostic mechanism among several ISPs, requiring a special server in each ISP that replies to diagnose queries from other domains.

[LOMZ07] represents a more closely related work in that it adapts a new measure of routing dynamics which weighs each AS link by the number of prefixes using that link, the same measure used in this chapter. However the scheme in [LOMZ07] uses the link weight changes observed from a *single* vantage point to construct an s-t graph; a min-cut on this graph represents the most likely set of edges where the routing changes originated. It only uses data from additional vantage points that is *relevant* to the primary point to resolve ambiguities in pinning down the change origins. In contrast, the design presented in this chapter utilizes data from all the monitors. The work by [FMM+04] can be considered as another closely related work in that it proposes a root cause inference system using data from multiple monitors simultaneously. The scheme aggregates BGP updates according to time, monitors and prefixes (in that order), that is, it derives root causes from analyzing *path* changes. Since multiple paths go through the same AS or AS links, this inherent coupling factor leads to the need of using a very large number of monitors to achieve adequate accuracy.

Principal Component Analysis has been applied to both traffic and routing information to help understand network dynamics. [LCD04b, LCD04a, LPC+04, LCD05] first proposed an approach based on PCA for detecting volume anoma-

lies in traffic data collected by several monitors within a network. A volume anomaly denotes unusual traffic load levels. They observed that, although traffic data is high-dimensional (in terms of number of links), normal traffic patterns are intrinsically low-dimensional. Thus, they separated network traffic into a normal subspace, and an anomalous subspace, and they used the minor components of PCA to identify volume anomalies. [RSRD07] showed that tuning PCA to operate effectively is nontrivial. [ZGGR05] introduced the *temporal PCA*, which exploits temporal correlation to identify dominant pattern across time. In contrast, [LCD04b, LCD04a, LPC+04, LCD05] analyzed the correlation between traffic on different links (*spacial PCA*).

The work that is most relevant to ours is [XCZ05] which applied PCA to routing data to analyze Internet-wide events. Given a stream of BGP updates collected by a *single* monitor over time, they group prefixes that are likely affected by the same network event. [HFLX07] focused on diagnosis of network disruptions within a single network, and used PCA to combine multiple BGP updates streams coming from distinct observation points. Both [XCZ05, HFLX07] used time-series matrices, which count the number of BGP updates received by a single router in each time slot. In contrast, our work exploits the topology dimension. For each given time slot, we measure the number of routing changes over each link or AS as observed by multiple monitors.

Our Contributions

In this chapter, we uses Principal Component Analysis to reduce dimensionality of the BGP routing data and to locate main contributors to routing dynamics. More specifically, we measure the number of routes going over each AS and each AS-AS link from the observed path changes, and abstract the routing changes as the change in the number of routes changing over each link and AS. By applying PCA to link and AS weight changes, we can identify the problematic spots that are involved with most routing changes.

We evaluate our approach through Internet scale simulations and BGP measurement data. The simulation results show that our approach can accurately pin down the failed links and ASes, is robust to simultaneous unrelated failures and can separate these unrelated events to distinct principle components. By applying our technique to routing data collected from multiple monitors, not only we identified instances of large routing changes, but also were able to clarify the scope of these instabilities by examining the degree of influence of individual monitors on the principal components. Different

from previous results in identify origins of instabilities by system and protocol heuristics [CGH03, CSK03, FMM+04, WMRW05], our approach represents a black-box design where we apply PCA, a well established statistical technique to the routing data collected from multiple vantage points. The large data set is difficult to analyze by following protocol rules to say the least, especially under the condition of unknown routing policies. However PCA enables us to understand the combined results of routing changes seen by different vantage points, and this whole process can be automated for systematical analysis.

3.2 Model and Methodology

We now define our model and discuss how we applied Principal Component Analysis to Internet routing data.

3.2.1 Contributors of Routing Dynamics

Whenever a BGP router changes its route to any destination prefix, it sends a BGP update to its neighbors.

In Figure 3.1, assume the link between 77-88 fails. As a result, AS 77 cannot use AS 88 as next hop to reach prefixes P1..P4. As a consequence, AS 77 starts using AS 99 to route to these prefixes and sends an update message to AS 55 and AS 66, communicating the new route to reach prefixes P1..P4 as shown in Figure 3.2. Similarly, AS 88 will also send updates to its neighbors since its route to AS 55 and AS 66 will not be valid after failure of 77-88. Since the BGP updates we see in the network in Figure 3.1 are originated by AS 77 and AS 88, we call them the *contributors* of the routing dynamics. Further, since a specific link between these two ASes is involved in all these BGP updates, we can say that the link 77-88 is a *contributor* of the routing dynamics in the network as well.

By examining the routing updates at monitors one can try to understand who is contributing the routing dynamics in the Internet. However inferring the contributors from even a single monitor is not straightforward. For example, in Figure 3.2, when observing the routes received at the collectors, we do not know the routing preferences of remote ASes and hence do not know which of the initial and final paths is the preferred path. The contributor could thus lie on either of the two paths. Add to this difficulty, the fact that different monitors can potentially see very different routing changes. Even for the same event, the effect on different monitors can be different. For example, in Figure 3.2, the same event can result in the two monitors AS 77 and AS 44 changing routes

3.2. Model and Methodology

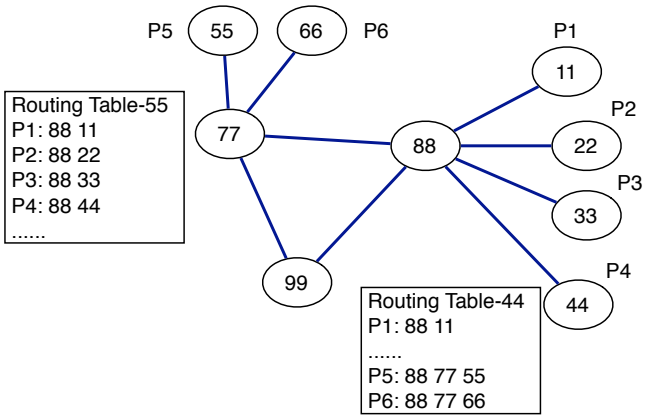


Figure 3.1: AS ii ($i \in \{1, 2, 3, 4, 5, 6\}$) announces prefix P$i$.

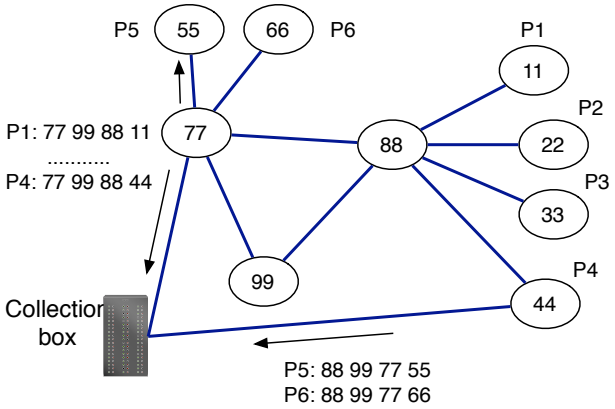


Figure 3.2: Route changes due to the fault of the link 77-88. AS 77 and AS 44 are connected to a collection box and act as monitors.

to different sets of prefixes; AS 77 sends updates for P1..P4, while AS 44 sends updates for P5, P6.

Hence understanding major contributors of routing dynamics is very chal-

lenging and aggregating changes using prefix level information may not yield the desired results. In this section, we introduce the metrics we use to capture routing dynamics and explain our PCA based approach.

3.2.2 AS and Link Metrics

To identify the contributors of routing dynamics, we need to extract the AS or AS link contributing dynamics from the observed AS path information. Given a prefix p and a monitor m, we call *route change* the change of the AS path that m uses to reach p. By examining both the initial and the final AS paths of a large amount of route changes affecting the Internet, one can expect to understand which ASes (or AS-links) possibly contributed these routing dynamics. Given a set of BGP updates around the same time, we assume that ASes (or AS-links) that appear more often in the paths (i.e. are affected by more route changes) are more likely to be originating the route changes. For the example in Figure 3.2, by examining the route changes we can see that 77-88 is the common link in the lost routes, while the common links in the new routes are 77-99 and 99-88.

To capture these common links, we define the metric *link-change* as the number of prefixes that see a particular AS-AS link in route changes from each monitor. When a prefix route changes, even the links from the old route are accounted for in this change. For example in Figure 3.2, from monitor 77 the link-change value of 77-88, 77-99 and 99-88 is 4, since 4 prefixes affected by route changes had these links in their initial or final paths.

Often however, there is no single link that is problematic, instead problems inside an AS result in route changes on its adjacent links. To capture the AS contributors, we define the metric *as-change* to refer to the number of prefixes that see a particular AS in route changes from each monitor. For example in Figure 3.2, the as-change of AS 77, AS 99, and AS 88 is 4, since 4 prefix routes affected by changes used these ASes in the initial or final path.

By using these two metrics, we hope to extract the most commonly occurring links and ASes in large scale routing changes seen from multiple monitors.

3.2.3 Constructing the PCA Input Matrix

Given a stream of BGP updates, we grouped them into time slots of T. We then constructed the PCA input matrix for each slot as follows. Let A be the set of all the Autonomous Systems, L be the set of all the links between Autonomous Systems ($L \subseteq A \times A$), and M be the set of all the monitors

($M \subseteq A$). For each time slot t, we constructed $X_A(t)$, a $a \times m$ matrix, where a is the number of ASes affected by any route change during the time bin t ($a \leq |A|$), and m is the number of monitors, which observed any route change during the time bin t ($m \leq |M|$). Each entry x_{ij} represents the as-change value indicating the number of prefixes that underwent a route change over the AS i observed by the monitor j during the time bin t. Likewise, we constructed $X_L(t)$, a $l \times m$ matrix, where l is the number of links affected by any route change during the time bin t ($l \leq |L|$). Each entry x_{ij} represents the link-change value indicating the number of prefixes that underwent a route change over the AS link i as observed by the monitor j during the time bin t. We converted both $X_A(t)$ and $X_B(t)$ to have zero mean, with the transformation $X = (X_{ij})$ with $X_{ij} = (X_{ij} - \mu_j)$, where μ_j is the mean of all the values of the column X_j. Thus, each entry x_{ij} indicates how much the observed number of prefix changes in the time bin j differs from the mean value observed by the same monitor. The two metrics of link-change and as-change can provide different insights into the event and in the remainder of the chapter we present both.

The value of T used to group updates needs to be large enough to reduce the likelihood that changes of a single event are split into multiple time slots. A value of $T \geq 10$ minutes is a reasonable choice and this value corresponds to a common choice in other prior work [HFLX07].

3.2.4 Analyzing Principal Components

After applying PCA on the matrix described above, we get a set of principal components and we analyze the ASes and AS links with reference to these components. We use the following procedure:

Principal components selection We first try to understand how many principal components we need to examine in order to explain most of the routing dynamics over a specific time interval. To achieve this, we select the principal components that account for most of the variance. In this chapter, we graphically show the variance of the principal components using the *scree plot*, i.e. a line plot of all the eigenvalues, sorted from high to the low. In Section 3.3, Figure 3.3(a) is an example of scree plot.

ASes and links analysis After identifying how many principal components we need, we project the ASes (and links) on these principal components in a plot called *score plot*. The first few principal components are meant

to capture the most variance and hence by projecting the AS (or link) set on the these principal components, one can expect that the *outlier* AS (or link) that stands apart from the rest is contributing a lot of routing changes. Figure 3.4(a) is an example of score plot of the first two components using the link-change metric where link 701-1239 shows up as an outlier.

Monitors analysis Finally, different monitors often see very different routing changes and we analyzed how monitors influence the scores of ASes (links), and influence the principal components. We do this by examining the projection of the monitors on the principal components in a plot called *load plot*. From this plot, we can get a better understanding of the scope of the route change(s) seen and can differentiate between local changes and global changes. Figure 3.4(c) shows a load plot where many monitors influence the first principal component, while Figure 3.8(b) is the load plot where only one monitor is influencing the first principal component.

3.3 Internet Scale Simulations

In this section, we use Internet scale simulations to understand how effective our approach is in detecting the major contributors of routing dynamics. Using an Internet scale topology we simulate routing events and for each event, we relate the results from PCA to the actual AS or link problems.

3.3.1 Setup and Route Computation

For our simulations, we use an AS topology inferred from BGP routing tables and updates, representing a snapshot of the Internet as of Feb 15 2006 (available from [irl]). The details of how this topology was constructed are described in [ZLMZ05]. The topology consists of 22,467 AS nodes and 63,883 links. We classified each link as either customer-provider or peer-peer using the PTE algorithm [Gao01] and used the *no valley prefer customer* routing policy to infer routing paths. We randomly picked 50 nodes and designate them as monitors. We can observe the routes from these monitors to all destinations.

We modeled each AS as a single node and used the routing tables collected from RV [roua] from the time of topology snapshot to get a mapping of how many prefixes were announced by each origin AS at that time. After this step we had a total of about 180,000 prefixes announced. We abstracted the

router decision process into the following priorities (1) local policy based on relationship, (2) AS path length, and (3) lowest ID tie-breaker. We applied our decision process to compute the routes from each monitor to all prefixes in the topology and record these routes as the initial set of routes. A similar setup and decision process was used in previous works such as [FMM+04]. Recent work [MFM+06] reported the inaccuracies in path predictions resulting from abstracting AS as a single node. However, such a setup is still useful for basic validation and evaluation since it can provide an important sanity check. Part of our future work is to extend the current setup to include the path prediction model from [MFM+06].

3.3.2 Routing Events

Once the initial set of routes was computed, our goal was to simulate various problem scenarios and see how accurately our method was in identifying the contributors of routing dynamics. We simulated three kinds of events described below.

Global link event We failed a single link between two tier-1 ASes. We picked the top-10 most used tier-1 links, and we failed each link individually in 10 distinct events.

Global AS event In our setup its not possible to simulate intra-AS problems, and removing a node from topology is too extreme. We observed from BGP data that during AS problems, multiple links adjacent to that AS are affected. Hence, we simulated problems with a major AS by failing 3 links adjacent to a tier-1 AS and recomputing the routes from all the monitors. We used a set of 8 popular tier-1 ASes for this event class and hence we had 8 such events.

Unrelated link events For each event in this category, we failed both a tier-1 link and a link adjacent or very close to one of the monitors. We picked those two links such that they did not have any common end points. The aim of simulating this class of failures is to see if our approach is robust enough to clearly separate local problems from global problems.

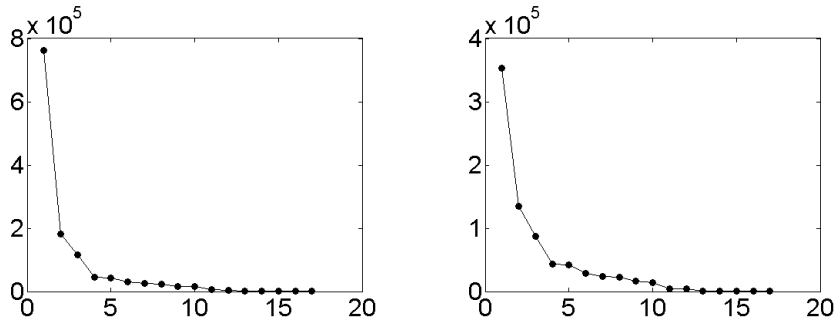
3.3.3 Results

Recall from Section 3.2 that we use two metrics - *link-change* and *as-change* - indicating the effects of route changes on a specific AS-AS link, or on a specific

AS. For each simulated event, we applied our approach on the routing changes observed from all the monitors, using both these metrics. For each category, we first present our approach applied on a specific representative case, and then describe the overall results of all the simulated events in the same category.

3.3.3.1 Understanding Global Link Events

As a representative we picked the failure of the link 1239-701. Figure 3.3(a) shows a scree plot representing the amount of variance captured by the principal components using the as-change metric. Figure 3.3(b) shows the scree plot using the link-change metric. We can see from these plots that the curve flattens out after first 3-5 principal components.



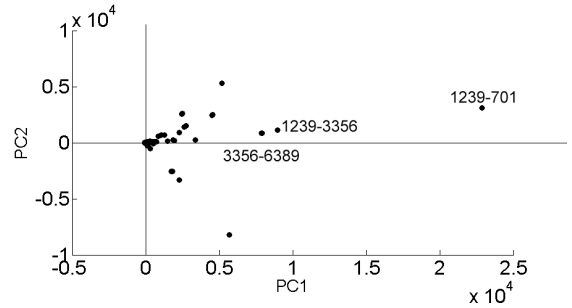
(a) as-change metric. (b) link-change metric.

Figure 3.3: Scree plot for global link event.

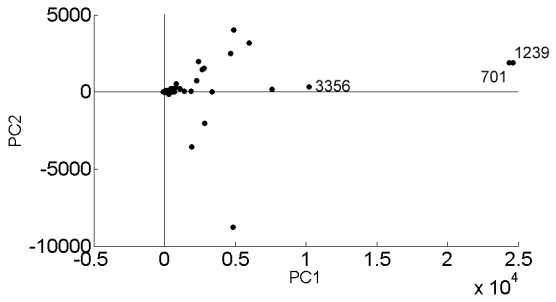
Figure 3.4(a) shows the score plot using the link-change metric. This plot represents a projection of AS links onto the first and second principal components wherein most of the variance lies. We can see that link 1239-701 clearly stands out from the rest. When using the as-change metric, the two endpoints of the link show up as main contributor ASes as shown in the score plot in Figure 3.4(b).

Further we examined the load plot using the link-change metric (Figure 3.4(c)), which shows how monitors influence the principal components. We can see that the monitors are well spread on the plot. Hence we can say that the route changes of most variant link on the first principal component are observed at multiple monitors.

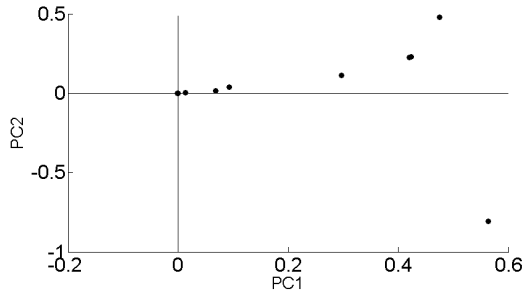
3.3. Internet Scale Simulations



(a) Score plot using link-change metric.



(b) Score plot using as-change metric.



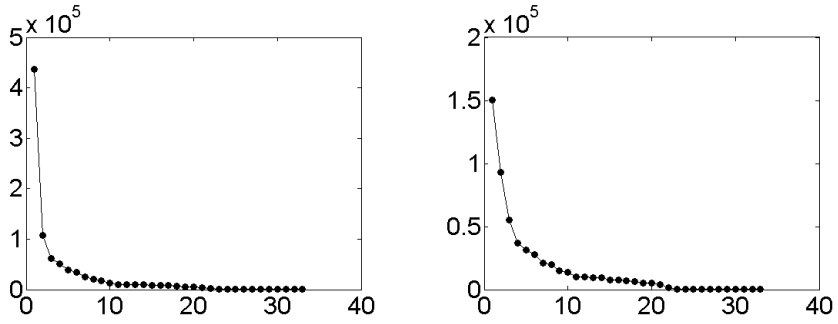
(c) Load plot using link-change metric.

Figure 3.4: Global link event.

Overall, on studying the results from simulations of global link problems, we found that in all the cases the failed link appears as the most variant link on the first principal component using the link-change metric. Likewise, with the as-change metric both the end points of the link always showed up as the most variant ASes on the first principal component.

3.3.3.2 Understanding Global AS Events

We first discuss a specific case of applying our approach on a simulated problem with AS 209 and then present overall results in this category. Figure 3.5(a) shows the scree plot representing the amount of variance captured by the principal components using as-change metric. Figure 3.5(b) shows the scree plot using the link-change metric. We can see from these plots that the curve starts to flatten out after the first 10 principal components. However, with link-change metric the 2nd, 3rd principal components contribute much more than in the case of as-change. Because multiple links have large changes, more principal components are needed to capture these changes using link-change metric, while since the simulated event mostly affects only one AS, lesser principal components are needed with the as-change metric.

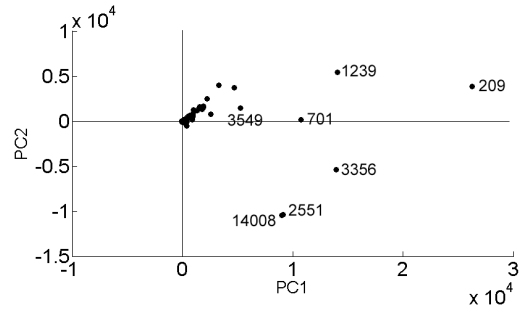


(a) as-change metric. (b) link-change metric.

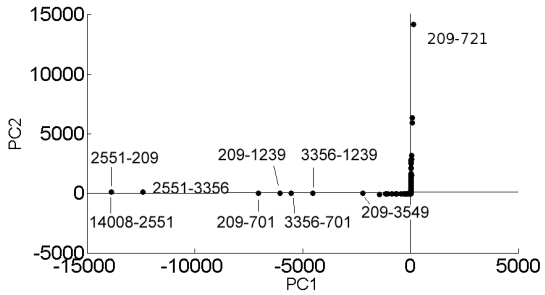
Figure 3.5: Scree plot for global AS events.

Figure 3.6(a) shows how the problem AS stands out in the score plot using the as-change metric. On the other hand, from the score plot on the link-change metric (Figure 3.6(b)), we can see that the failed links appear at different locations. Further the the most variant link on the first principal component

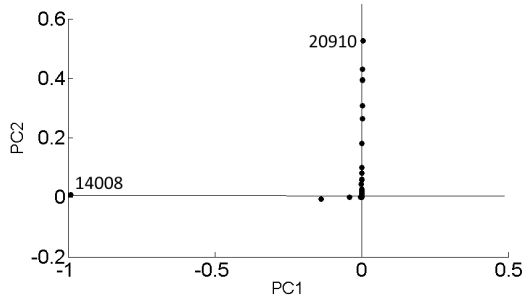
3.3. Internet Scale Simulations



(a) Score plot using as-change metric.



(b) Score plot using link-change metric.



(c) Load plot using link-change metric.

Figure 3.6: Global AS event.

i.e. 2551-209 is not the failed link. When multiple links out of AS 209 failed, the aggregated effect was observed over the link 2551-209 from one of the monitors, and as shown in the load plot in Figure 3.6(c) this monitor is influencing the first principal component. If we remove this outlier monitor and reapply PCA, we see a better distribution of the monitors in the load plot and cleaner results with the link-change metric.

Overall, on analyzing the results from simulations of global AS problems, we found that in all the cases, the problematic AS showed up as the most variant on the first principal component using the as-change metric.

Table 3.1 summarizes the overall results using the link-change metric. Each column indicates a particular simulation event, while each row indicates the links failed in that simulation. A value of x in cell (i, j) indicates that for event j, the highest rank of the link i on the top 6 principal components is x. Thus, lower the value, better the chances of link standing out from the rest. Table 3.1 shows that in 9 over 10 case, all three failed links appear in the first 6 positions of the first 6 principal components.

	events							
	I	II	III	IV	V	VI	VII	VIII
link1	3	1	1	2	1	4	1	1
link2	2	2	2	1	1	3	3	1
link3	5	5	1	4	1	6	6	1

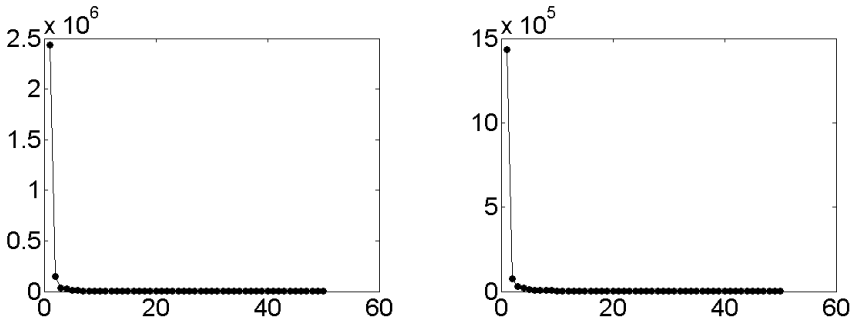
Table 3.1: Overall results for global AS events.

3.3.3.3 Understanding Unrelated Link Events

Finally, we looked at events involving failures of two unrelated links at the same time. We first present a representative case of failures of links 1239-701 and 17759-9304, and then summarize overall results. Figure 3.7(a) and Figure 3.7(b) shows the scree plots using as-change and link-change metrics. The top variance is much higher in this case and hence the curves look different from previously presented scree plots.

Figure 3.8(a) shows the score plot showing the projection of AS links on the first two principal components. We can observe that the two failed links appear as the two most active links on the first and second principal components.

To understand how the monitors influence the principal components, we examined Figure 3.8(b) showing the load plot for the link-change metric. We



(a) as-change metric. (b) link-change metric.

Figure 3.7: Scree plot for simultaneous unrelated link events.

can see that one monitor is strongly influencing the first principal component (local event), while the second principal component is influenced by a wider set of monitors (global event).

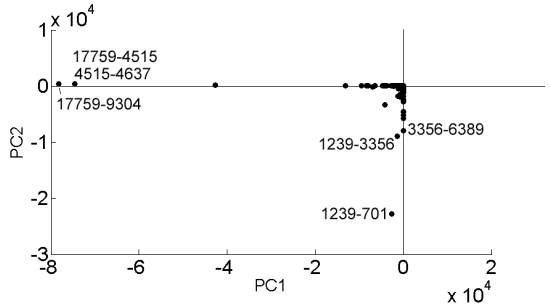
Table 3.2 summarizes the overall results for simultaneous unrelated link problems simulations. Table 3.2 shows that in 8 over 10 cases both the failed links appear as the most variant in one of the first 6 principal components, while in 2 cases, one of the failed link shows up in the 2nd and 5th position.

Summarising, the simulation results show that the as-change metric helps in accurate identification of large scale AS events, while the link-change metric works more effectively for link problems. We saw that it is important to understand how the monitors influence the principal components in order to understand the results. Further, in the presence of multiple events, the most active contributors appear on separate principal components.

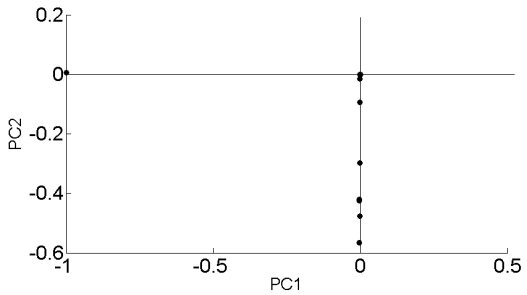
3.4 Measurements

In this section, we report the results from applying our method to the routing data collected by RouteViews and RIS. We pick a subset of 38 monitors (11 from RIS and 27 from RouteViews) which are not topologically adjacent of each other (as observed from routing tables) in order to have a diverse view of the Internet. We analyze BGP routing data for the week from May 10, 2007 to May 16, 2007 and for the week from June 1, 2007 to June 7, 2007. The results

3. Identifying Contributors of Routing Dynamics using Multiple Views



(a) Score plot using link-change metric.



(b) Load plot using link-change metric.

Figure 3.8: Simultaneous unrelated link events.

	events									
	I	II	III	IV	V	VI	VII	VIII	IX	X
link1	1	1	1	1	1	1	1	1	1	1
link2	1	1	1	2	5	1	1	1	1	1

Table 3.2: Overall results for simultaneous unrelated link events.

from the two weeks are very similar, thus we report the results for the week of May only. In the remainder of the section, we use the term *reference week* to refer to the week in May, and the term *all monitors* to refer to the 38 selected monitors.

3.4.1 Routing Dynamics by Different Measurements

We divide one week data set into one hour bins, and Figure 3.9 shows the total number of updates collected every hour by all monitors during the reference week.

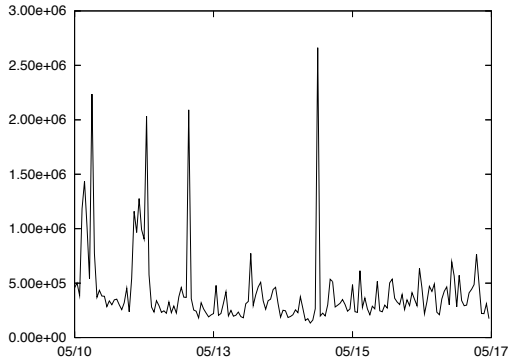


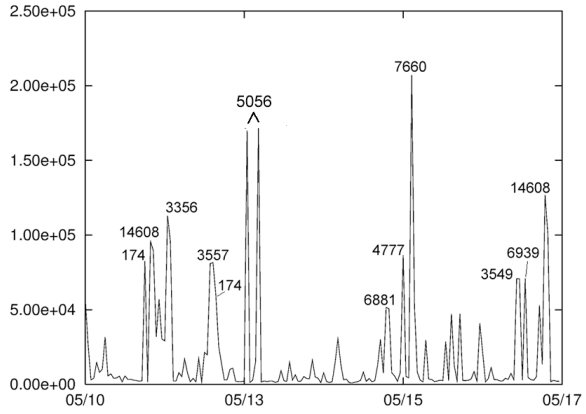
Figure 3.9: Update counts over 1 week period.

To get an idea of whether any major routing changes occurred during the reference week, for each time bin we calculate the variance of the most active AS on the first principal component, and plot these values over time (Figure 3.10(a)). Likewise, Figure 3.10(b) shows the highest variance for a link on the first principal component in each of the one hour intervals. We label the top few spikes of ASes and links having the highest variance. Both Figure 3.10(a) and Figure 3.10(b) show evidence of large scale routing dynamics over the reference week. Comparing Figure 3.9 with Figures 3.10(a) and Figures 3.10(b), we see that the highest spikes in the update count does not necessarily correspond to high spikes in the variance plots, and vice versa.

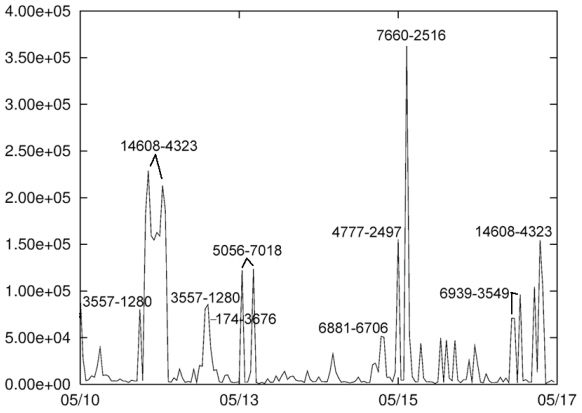
To understand routing dynamics behind the highest spikes in Figures 3.10(a), and Figures 3.10(b), we applied our method to the time intervals when these spikes appear. Next, we describe two representative cases of events that caused spikes.

3.4.2 Case I: Routing Dynamics Due to a Local Event

First we analyse the highest spike in Figure 3.10(a) attributed to the AS 7660, occurring on May 14, 2007 during the time interval 19:00-20:00. At the same



(a) as-change metric.



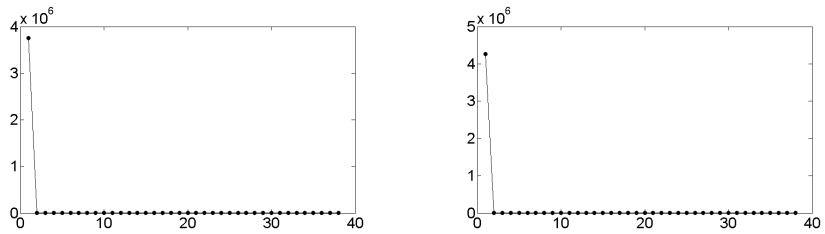
(b) link-change metric.

Figure 3.10: Highest variance of first principal component over 1 week period.

time Figure 3.10(a) presents a high spike for the link 7660-2516.

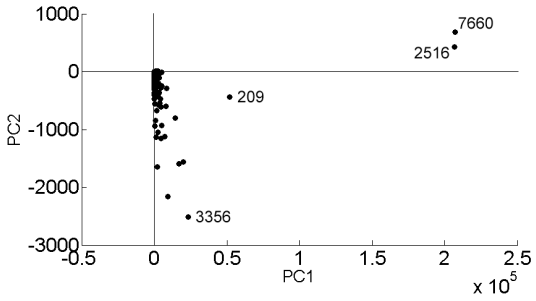
Looking at the scree plots for the as-change metric (Figure 3.11(a)) and for the link-change metric (Figure 3.11(b)), we observe that the first principal component explains most of the variance. Hence, we can examine only the first two principal components nearly without any loss of information.

3.4. Measurements

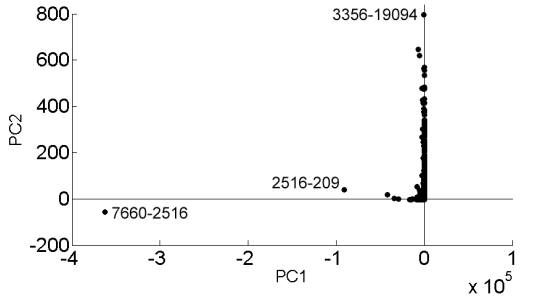


(a) as-change metric. (b) link-change metric.

Figure 3.11: Scree plot for case I.



(a) as-change metric.



(b) link-change metric.

Figure 3.12: Score plots for case I.

The score plot for the as-change metric in Figure 3.12(a) presents AS 7660 and AS 2516 as the outliers of the routing dynamics. Figure 3.12(b) shows that 7660-2516 is by far the link mostly involved in route changes.

By examining the load plot in Figure 3.13, we see that the monitor 7660 stands out. This means that the monitor 7660 contributes most heavily to the first principal component values. We then remove this monitor from the data set and reapply our method.

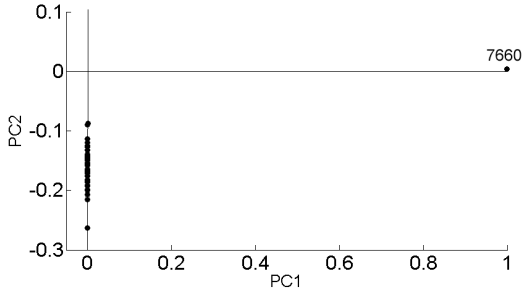


Figure 3.13: Load plot for case I (as-change metric).

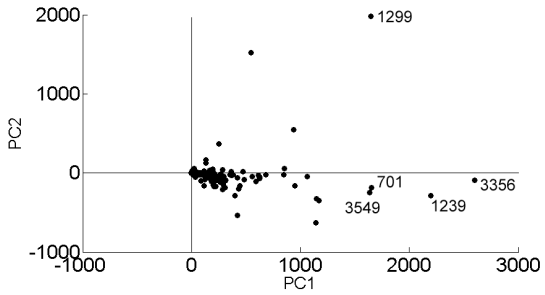
We observe that the resulting new load plot (Figure 3.14(b)) shows no major outlier, and the new score plots for the as-change metric in Figure 3.14(a) presents a much reduced variance. AS 3356, which was on the highest position of the second principal component before removing the monitor 7660, now it stands out as the most active on the first component. However one should note that the value of PC1 axis in Figure 3.14(b) is 2-order of magnitude smaller than that in Figure 3.11(a). We can conclude that the observed routing changes during the time interval analysed in this case study is due a local event over the link between AS7660 and 2516, which is observed by the monitor 7660 mostly.

The spikes in Figure 3.10(b) and Figure 3.10(b) involving, respectively, link 14608-4323 and AS 14608 have a similar nature, since the principal components are mostly influenced by a single monitor.

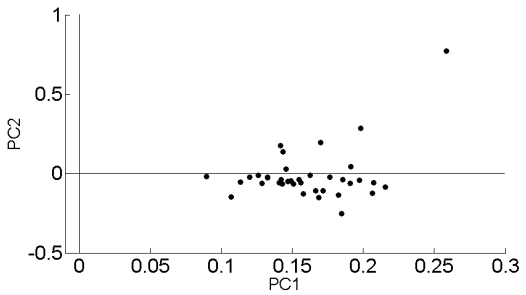
3.4.3 Case II: Routing Dynamics Due to a Global Event

Next we study the spike in Figure 3.10(a) on May 11, 2007 22:00-23:00, corresponding to AS 174. Note that, at the same time, Figure 3.10(b) shows a minor spike for the link 174-3676.

3.4. Measurements



(a) Score plot with as-change metric.



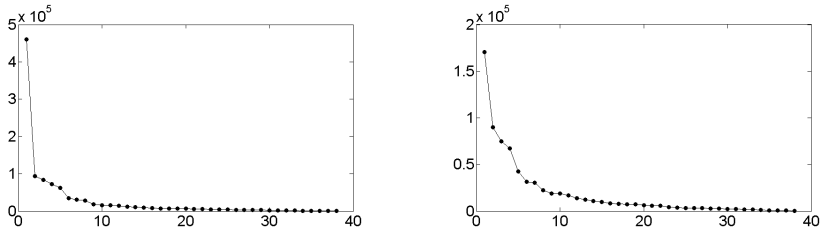
(b) Load plot with as-change metric.

Figure 3.14: Case I after removing monitor 7660.

We first look at scree plot that gives us an idea of how many components we need to examine to correctly explain the ongoing routing dynamics. Figure 3.15(a) shows that, with the as-change metric, the first principal component captures about 80% of the variance, while the first 5 components account for over 90%. Interestingly, according to Figure 3.15(b), we need to take into account a lot more components when we use the link-change metric.

Plotting the scores of the first and second principal components for the as-change metric (Figure 3.16(a)), we see that AS 174 clearly stands out from the others. From the load plot using as-change (not shown for lack of space), we saw that a bunch of monitors collectively influenced the first principal component, which indicates that the problem with AS 174 was not local.

Figure 3.16(b) represents the score plot on the first two components using



(a) as-change metric. (b) link-change metric.

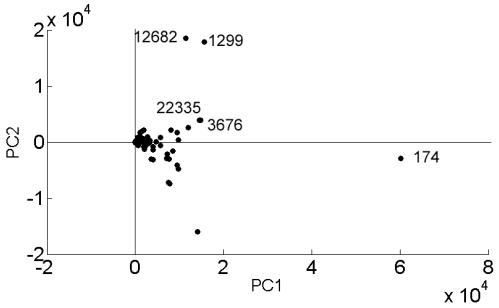
Figure 3.15: Scree plot for case II.

the link-change metric over the same time interval. In this plot, the links 174-3676 and 12682-174 are the most active on, respectively, the first and the second component. One can also see that the other links in the top 3 on first principal component are adjacent to each other and the outlier link and hence most likely appear as an effect of problem with the outlier link and such analysis can be performed to put the positions of links in perspective. Further, Figure 3.16(c) indicates that the first principal component is not influenced by any single monitor alone, while the second principal component is mainly influenced by monitor AS 12682 which is directly adjacent to AS 174. Other links adjacent to AS 174 appear in highest positions of other minor principal components as well. This justifies why we may need to look at many components to understand the route changes.

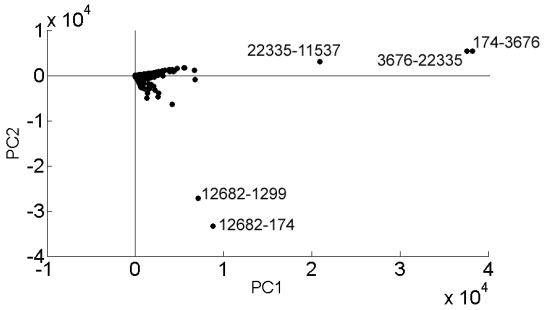
We can conclude that the large scale routing changes analysed in this case study are most likely related to problems within AS 174, which result in routing dynamics affecting many ASes peering with AS 174. On checking NANOG mailing lists, we found some messages indicating network issues with Cogent (AS174) during the same time period, which confirms our understanding of the event.

Summarising, the analysis of real world case studies shows that our method effectively applies to real world scenarios, as well as to a simulated environment. It provides us a way to understand large routing events, separating major routing dynamics from minor events. Using our method, we can identify time periods of large scale routing changes, pinpoint ASes and links mostly involved, and understand the impact of routing events comparing views of different monitors.

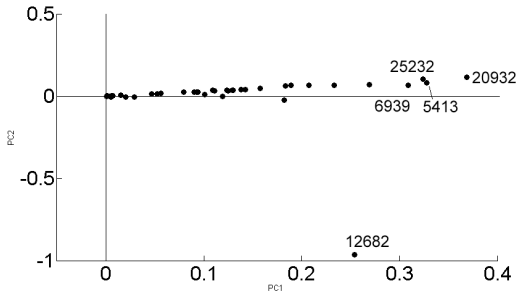
3.4. Measurements



(a) Score plot for case II (as-change metric).



(b) Score plot for case II (link-change metric).



(c) Load plot for case II (link-change metric).

Figure 3.16: Case II

3.5 Conclusions

Due to complex interconnects among ASes and private routing policies, different vantage points in the Internet routing infrastructure observe different routings changes, creating a challenge in utilizing data from multiple vantage points to pin down the locations of most problematic spots.

In this chapter we first define a new metrics of as-change and link-change to capture routing changes, and apply PCA to these metrics observed from multiple vantage points. Our new measure helps pin down the changes to specific ASes or links, and is advantageous over the previous approach of using paths changes since one failure can affect large numbers of paths. We evaluated our approach through simulations and real case studies. The results show that we can accurately locate the problematic AS or AS link that contributes to observed routing changes. In addition, given the fact that different vantage points see different views, the application of PCA enables us to separate unrelated routing events to different principal components.

Chapter 4

Extracting Inter-AS Peerings from the Internet Routing Registry

Interdomain routing policies of many of the networks on Internet are documented in the Internet Routing Registry. We propose a novel methodology to extract peering information from the Internet Routing Registry and we provide an implementation as an online service. Both the method and the service are based on: a consistency manager for integrating information across different registries, an RPSL analyzer that extracts peering specifications from RPSL objects, and a peering classifier that aims at understanding to what extent such peering specifications actually contribute to fully determine a peering. A peering graph is built with different levels of confidence. We compare the effectiveness of our approach with the state of the art.

The main results presented in this chapter are also described in [DRR06].

The chapter is organized as follows. Section4.1 describes previous work and summarizes our contributions. In Sections 4.3 4.4 we present our methodology and the system we developed to extract peering information from RPSL data. We use our system to analyze the data set specified in Section 4.2. A quantitative comparison with the state of the art is done in Section 4.5. Conclusions are summarized in Section 4.6.

4.1 Introduction

The *Internet Routing Registry* (*IRR*) [ripc, irra] is a large distributed repository of information, containing the inter-domain routing policies of many of the networks that compose the Internet. The IRR was established in 1995 with the main purpose to promote stability, consistency, and security of the global Internet routing. It consists of several registries that are maintained on a voluntary basis. The IRR can be used by operators to look up peering agreements, to study optimal policies, and to (possibly automatically) configure routers.

Previous Work

There is a wide discussion about the current role of the IRR [SF04]. Some people do not consider it very useful, because it contains a lot of outdated data. Others have put in evidence its importance to understand the Internet routing, because it contains unique and significant information. Anyway, it is undeniable that the IRR keeps on being fed by many operators, that several tools have been developed to deal with the IRR (see, e.g., IRRToolSet [irrc]), and that there are research issues on the Internet routing, at least partially, based on the content of the IRR. However, as pointed out in [SF04], extracting information from the IRR is far from trivial: the policies written in RPSL can be quite complex, the level of accuracy of the descriptions largely varies, and, also because of its distributed nature, the IRR contains many inconsistencies [Ker02].

For example, the RIPE offers an IRR consistency check service (RRCC) [SGK+01, rrc] that aims at detecting unregistered peerings. It verifies whether a peering that can be inferred from operational routing data is also described, in some form, into the IRR. We will show later that currently the RRCC extracts peerings from the IRR in a way that is much less accurate than the one presented in this chapter. Actually, the need of a better analysis of the content of the IRR is pointed out by the RIPE itself that considers this as a long term goal [rrc].

On the research side, Mahadevan et al. [MKF+06] presented a comparison of several characteristics of the AS-level topologies built on the basis of different data sources, including the IRR. They also proposed a metric to characterize such topologies. Zhang et al. [ZLMZ05] derived an AS-level topology combining IRR data with BGP routing information collected from multiple sources, such as RouteViews [roua], looking glasses, and route servers. They showed that the data from the RIPE routing registry reveal topology information which cannot be found in other sources. Siganos et al. [SF04] developed a tool, called

Nemecis [nem], that checks the correctness of IRR data and their consistency with respect to BGP routing table information. They argued that 28% of ASes have both correct and consistent policies and that the RIPE routing registry is by far the most accurate. Carmignani et al. [CBD+02] presented a service for the visualization of IRR data. We shall compare the level of accuracy of the methods for extracting peerings from the IRR used in the above papers with respect to ours.

Our Contributions

The purpose of this chapter is to describe an on-line service and its underlying methodology, that extracts peering information from the IRR. We believe that our service can have beneficial effects both for operators and for several research projects. The main results presented in this chapter can be summarized as follows.

- We describe a method and a on-line service to extract peering relationships from the IRR. Both the method and the service are based on: a consistency manager for integrating information across different registries, an RPSL analyzer that extracts peering specifications from RPSL objects, and a peering classifier that aims at understanding to what extent such peering specifications actually contribute to fully determine a peering. A peering graph is built with different levels of confidence.
- We prove the effectiveness of our method by showing that it allows to discover many more peerings than the state of the art.
- We provide an implementation of our method as an on-line service, available at `http://tocai.dia.uniroma3.it/~irr_analysis`.
- As a side effect, our study highlights how the different RPSL constructions are actually used to specify peerings.

4.2 Our Dataset

The registry data we use throughout this chapter has been downloaded from [ripb, rad] on 03/31/2006. At that time there were 68 routing registries available for download, which are listed in Table 4.1. The registries are sorted according to their size in terms of number of `aut-num` objects registered inside them (2nd column). Void registries are omitted.

ripe	11468	92%	host	10	90%	reach	2	50%
apnic	3299	84%	ottix	9	33%	nestegg	2	100%
radb	2695	77%	csas	9	100%	gw	2	100%
arin	555	41%	rogers	8	100%	bendtel	2	50%
verio	498	42%	risq	8	100%	univali	1	100%
dodnic	254	11%	crc	8	62%	soundinternet	1	100%
altdb	249	63%	deru	7	0%	panix	1	0%
savvis	180	75%	sprint	6	16%	openface	1	100%
epoch	137	100%	bcnet	5	60%	koren	1	100%
level3	126	40%	vdn	4	25%	gts	1	100%
bell	74	98%	rgnet	4	100%	gt	1	100%
aoltw	53	3%	mto	4	25%	fastvibe	1	100%
jpirr	43	34%	easynet	4	100%	eicat	1	100%
sinet	28	10%	digitalrealm	4	100%	ebit	1	100%
arcstar	16	6%	look	3	100%	area151	1	100%
chtr	11	0%	retina	2	50%			

Table 4.1: `aut-num` in the routing registries before and after resolving inter-registry inconsistencies.

Table 4.2 indicates the level of overlapping between the largest registries. For each pair of registries (R_{i_1}, R_{i_2}) the table provides the number of `aut-num` objects that are registered both in R_{i_1} and in R_{i_2}. The main diagonal (R_i, R_i) reports the count of `aut-num`s appearing in registry R_i only.

	apnic	arin	radb	ripe	verio
apnic	2688	1	423	19	113
arin	1	463	37	7	14
radb	423	37	2037	50	45
ripe	19	7	50	11238	23
verio	113	14	45	23	310

Table 4.2: Overlapping `aut-num`s between registries.

Figure 4.1 gives an idea of the amount of work of the operators on the IRR over time. Namely, it shows the daily percentage of size variation of the RIPE routing registry (that is by far the most popular) over the period 11/14/05–

04/26/06. The plot shows that the RIPE registry keeps on being updated on a regular basis and that it grows of about 2% per month. Note that, compared to the RIPE registry, the update rate of the other registries is negligible. Our reference date (arrow in the plot) has been selected to be one with an average number of updates.

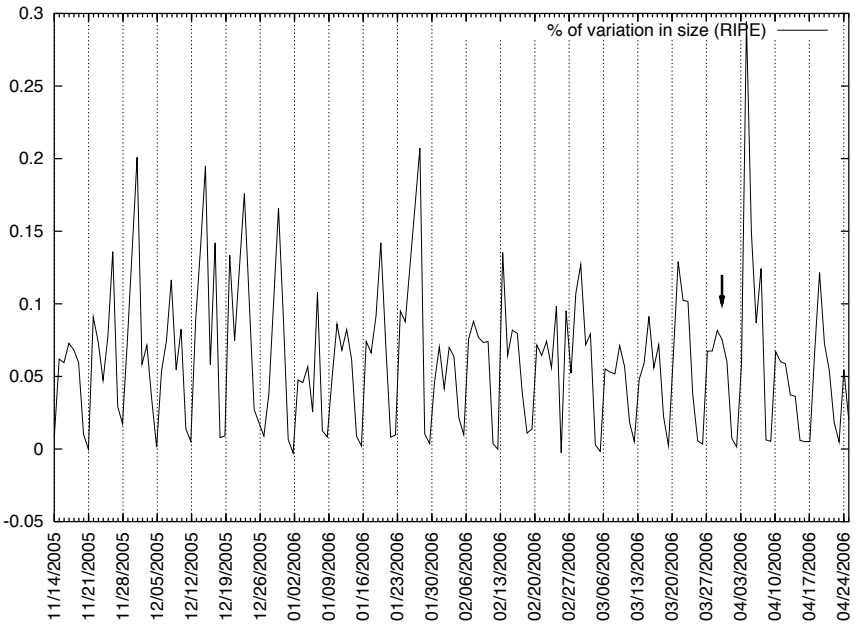


Figure 4.1: Daily growth of the RIPE routing registry.

4.3 RPSL Analysis Service: a Peering Extraction Tool

Our method for extracting peerings has been implemented and is available as an on-line service at `http://tocai.dia.uniroma3.it/~irr_analysis`. The service produces, on a daily basis: (i) General statistics on the IRR (number of objects defined in each registry, amount of overlapping information between registries, etc.). (ii) A set of pairs of ASes, corresponding to peering relation-

ships extracted from the IRR. Each pair is labeled with information about the context where it has been found, like the type of policy and the registry. The architecture of the service is composed by the following main blocks.

Basic Info Registry Analyzer provides preliminary information on the registries. For example, it computes the number of `aut-num`s and `as-set`s inside each registry. Also, it computes the "amount of overlap" between pairs of registries. Further, it deals with the evolution over time of registries, measuring the number of everyday updates. Such basic information is useful for giving a correct interpretation of the results obtained by using the service.

Inter-Registry Consistency Manager starting from a set of registries that, considered as a whole, may contain inconsistent information, constructs a purged new consistent version of the IRR. RPSL objects with the same key appearing in different registries are compared. A choice is done relying on the timestamp of the last change and in terms of the semantics of the attributes.

RPSL Peering Specification Analyzer extracts from the IRR the peering relationships between ASes. This is done by analyzing the body of RPSL objects. The relationships extracted in this phase are *candidate peerings* for the subsequent elaboration. In this step we also evaluate the current usage of the RPSL syntax constructions for expressing peerings. This block exploits IRRd and Peval.

Peering Classifier classifies the computed candidate peerings according to their relative matchings in order to understand to what extent they contribute to fully specify a peering. The output of this step is a peering graph, that can be constructed with different levels of confidence.

The above blocks will be detailed in the following sections.

4.4 How to Extract BGP Peering Information from the Internet Routing Registry

4.4.1 Integrating Registries

RFC 2622 [AVG+99] considers the IRR system as a whole. However, the IRR is composed by several registries, and the same object may be defined in many

of them. For example, in our data set, AS2510 is registered both in APNIC and in JPIRR. Of course, the presence of multiple definitions of the same object can lead to inconsistencies. Our Inter-Registry Consistency Manager takes care of resolving them. It takes as input a set of registries and processes them in order to build a new repository where each RPSL object is defined only once. Whenever it detects for a certain RPSL object the presence of multiple definitions (possibly coming from different registries), it examines all the definitions in order to determine which of them contains the most significant information. Such definition is kept in the final repository, while all the others are discarded. In what follows, a triple (x; y; z) represents a number of `aut-num`s, `as-set`s, `peering-set`s, respectively. In our data set we have (19,800; 7,798; 149) overall definitions. Among them, (18,735; 7,478; 149) are unique. Hence, potential inconsistencies affect at most (1,065; 300; 0) objects.

If an RPSL object is defined multiple times, the most informative definition is selected. We call *stub object* an `aut-num` object which misses information about BGP policies or a `set` object which misses the specification of the set members (consider that some attributes of RPSL objects are optional). Operators sometimes use stub objects as "placeholders" which can be referred to inside other parties' BGP policies. Our data set contains (3,133; 206; 11) stub definitions. If we detect that an object appears in more than one registry, we discard its stub instances. Since stub objects do not provide useful data about the existence of peerings, this does not cause any loss of information.

However, it may still be the case that several registries contain non-stub instances of a single RPSL object. If this happens, we select the instance with the most recent update timestamp, that is contained in the `changed` attribute. After removing the stub definitions and selecting the most recent timestamp, the potential inconsistencies affect at most (44; 77; 0) objects. Note that, even if the `changed` attribute is optional, in our data set there is only one definition that misses the timestamp over 2,271,446 objects in the IRR.

Yet, if there are (at least) two instances with the same most recent date, we select the definition belonging to the registry with highest *rank*. We rank the registries according to their size. This choice is somehow arbitrary. However, a registry with a higher number of objects often provides more reliable information than the others. Also, as shown above, the choice impacts very few objects. Last, we have inspected the objects that have multiple definitions with the most recent date and discovered that in most cases their definitions coincide. Of course, other rankings could be applied without impacting the general structure of the method.

The third column of Table 4.1 shows the percentage of the remaining

`aut-num` objects per registry after running the Inter-Registry Consistency Manager. It is interesting to observe that the RIPE registry has the highest absolute number and the highest percentage among the top 5 registries.

4.4.2 Discovering Peerings Through RPSL Analysis

In this section we detail the procedure we apply to extract peering information from RPSL data. As already stated in Section 1.2.2, peering specifications only appear in the `[mp-]import`, `[mp-]export`, and `[mp-]default` attributes of `aut-num` objects. Hence, `aut-num`s are the starting points of the peering extraction.

What follows is a fragment (25 lines, `ASX1-ASX13` represent ASes) of RPSL code that puts in evidence many of the problems encountered while discovering peerings in the IRR. We now show how to extract from this fragment the peerers of `ASX5`.

```
1.  peering-set: ASX1:PRNG-Y1  4. peering-set: PRNG-Y2
2.  peering: PRNG-Y2           5. peering: ASX7
3.  peering: ASX6

6.  as-set: ASX1:AS-Z1         9.  as-set: ASX2:AS-Z2
7.  members: ASX8, ASX9        10. members: ASX2, ASX4
8.  mbrs-by-ref: MNTR-ASX1

11. aut-num: ASX10
12. member-of: ASX1:AS-Z1
13. mnt-by: MNTR-ASX1

14. aut-num: ASX5
15. import: { from ASX2:AS-Z2 accept 100.0.0.0/8;
16.         } refine {
17.           from ASX1 ASX2 accept 100.1.0.0/16;
18.         } except {
19.           from ASX3 accept 100.1.1.0/24;}
20. export: to ASX1:PRNG-Y1
21.         to ASX1:AS-Z1 except ASX9
22.         announce 100.1.1.0/24
23. mp-export: to ASX11 at 2001::1 announce 2001::/48
24. default: to ASX12 action pref=10
```

```
25. default: to ASX13 100.1.1.1 at 100.1.1.2
```

By scanning such a code with the RRCC scripts [rrc], the following peerers are found: `ASX1`, `ASX3`, `ASX12`, `ASX13`. They come out by examining the lines 17, 19 (`import from`), 24, 25 (`default to`). However, such peerings are neither correct nor complete. On one hand, the peering between `ASX5` and `ASX1` does not hold, since the `refine` semantics require to compute the intersection between `ASX2:AS-Z2` and ASes `ASX1`, `ASX2`. On the other hand, there are peerers of `ASX5` that have not been discovered. The peerers `ASX6` and `ASX7` can be inferred only by considering all the ASes that belong to the `peering-set` used at line 20 and defined at lines 1-5. Further, the peerers `ASX8` and `ASX10` can be inferred only by considering all the ASes that belong to the `as-set` used at line 21 and defined at lines 6-8,11-13. Finally, the peerers `ASX2` and `ASX11` are not discovered because the scripts in [rrc] support neither multiple peerings appearing in the same `from` expression, nor the `mp-export` attribute. Even if the example is not taken from the real life IRR, it is a patchwork of pieces of code that are quite common in RPSL objects.

We now show our method for extracting peerings from the RPSL code. We describe how we build a set of candidate peerings, which we use later to identify peerings.

For each `aut-num` object A we compute three sets $import(A)$, $export(A)$, and $default(A)$ of candidate peerers corresponding to the `[mp-]import`, `[mp-]export`, and `[mp-]default` attributes, respectively. We describe our procedure with reference to the `import` attributes. The other attributes are processed in a similar way. If A defines a private AS, it is discarded, because it should not be visible in the Internet.

An `import` attribute may contain a simple or a structured policy. A simple `import` policy may contain several peering specifications. In this case $import(A)$ is the union of the candidate peerers corresponding to such peering specifications. If a peering specification is a `peering-set`, it is recursively expanded into its members. If it involves information about routers (e.g., interfaces, `inet-rtr`s, `rtr-set`s), they are removed. We keep only AS names, `as-set` names, set operators, and the keyword `AS-ANY`. The resulting expression is evaluated by using Peval. The output of Peval, consisting of a set of ASes, contributes to the set of candidate peerers.

A structured policy is a policy that has `except` and/or `refine` operators. In this case, $import(A)$ is still the union of several candidate peerers, but such candidate peerers are determined in a different way. First, we extract the two arguments of the `except` (`refine`) operator, which are simple policies. Then,

we process such policies as above, thus obtaining two sets of peerers. The union (intersection) of these two sets is our set of candidate peerers. If there are multiple `except`/`refine` expressions, we process them iteratively.

If an `aut-num` has many `import` attributes the above procedure is repeated for each one.

Finally, private ASes in $import(A)$ are removed.

Some technical issues should be pointed out. For example, a peering specification may contain the `AS-ANY` keyword. `AS-ANY` is either used "alone" (e.g. `import .... from AS-ANY ....`) or in a structured policy. In the first case one could argue that there is an AS that has a peering with all the other ASes, which is clearly unrealistic. Hence, in this case we discard the peering specification. Else, if `AS-ANY` is used inside a structured policy, we apply the above algorithm. Last, observe that also `inet-rtr` objects may contain information about peerings. However, we do not consider such peerings meaningful unless they appear in an `[mp-]import`, `[mp-]export`, or `[mp-]default` attribute of an `aut-num`.

Table 4.3 shows the incidence of RPSL constructions in the specification of peerings.

Table 4.4 shows the number of peering candidates extracted from the registries.

4.4.3 Constructing a Peering Graph

Once a peering candidate has been extracted from the IRR, it is classified according to the following two categories. Let A and B be the two ASes participating in the peering candidate. $A \xrightarrow{E} B$ represents the fact that A registered an export policy allowing BGP announcements to be sent to B. In turn, $A \xrightarrow{I} B$ indicates that B registered a policy according to which B accepts incoming announcements from A. The peering candidates are also tagged with the registries from which they have been extracted.

At this point, the peering candidates are used to determine whether there actually is a peering between two ASes. For example if, for two ASes A and B, we have found four peering candidates of type $A \xrightarrow{E} B$, $A \xrightarrow{I} B$, $A \xleftarrow{E} B$, $A \xleftarrow{I} B$, it means that both A and B have fully considered their partner in the peering. Hence, we call this peering *full peering* (A—B). Of course, there can be cases when the policies describe a peering only partially. For example, we might have only $A \xrightarrow{E} B$, $A \xrightarrow{I} B$, in which case the announcements from A to B are described in the policies, while there is no evidence of policies allowing

4.4. How to Extract BGP Peering Information from the Internet Routing Registry

`aut-num` objects	Action	Uses Peval	Occurrences
having a `default` attribute	Supported	No	4,851
having an `mp-import`, an `mp-export`, or an `mp-default` attribute	Supported	No	220
having a `peering-set` object in (*)	Supported	No	16
having an `as-set` object in (*)	Supported	Yes	939
having AS-ANY in (*) without further specifications	Discarded	No	660
having AS-ANY in (*) within a `refine` expression	Supported	No	24
having an `and`, an `or`, a `not`, or an `except` operator in (*)	Supported	Yes	5
having a `refine` or `except` expression in (*)	Supported	No	29
registering a private AS	Discarded	No	1
Private ASes in (*)	Discarded	No	86
`inet-rtr` objects having `peer` attributes	Discarded	No	217

Table 4.3: Incidence of different RPSL constructions in the specification of peerings. (*): an `[mp-]import`, an `[mp-]export`, or a `[mp-]default` policy.

ripe	342995	bell	974	risq	67	look	16	soundinternet	8
verio	118999	fastvibe	968	sinet	50	eicat	15	gw	8
radb	19309	level3	558	ottix	38	nestegg	14	digitalrealm	8
apnic	13979	epoch	439	jpirr	38	mto	14	univali	6
reach	9402	dodnic	389	csas	36	area151	14	gts	2
savvis	1593	gt	219	retina	22	openface	10	easynet	2
arin	1233	rogers	134	crc	22	bendtel	10	aoltw	2
altdb	1068	host	79	bcnet	18				

Table 4.4: Peering candidates per routing registry.

announcements from B to A. We call this situation *half peering* ($A \overset{1/2}{—} B$).

Table 4.5 shows all the possible relationships between two ASes. The col-

umn Peering Type associates a symbol to each possible situation. The column # of Peerings counts the peerings of each category. The column Single Registry reports the percentage of cases where all the candidate peerings contributing to the peering are in a single registry. We detail such percentage for the RIPE registry. A self peering refers to an AS that registers a peering with itself.

The peering types of Table 4.5 can be used to construct Internet topologies with different levels of confidence.

Policy Type				**Peering Type**	**# of Peerings**	**Single Registry**	**RIPE Only**
$A \overset{E}{\rightarrow} B$	$A \overset{I}{\rightarrow} B$	$A \overset{E}{\leftarrow} B$	$A \overset{I}{\leftarrow} B$				
√	√	√	√	$A \text{ —— } B$	42,599	96,7%	94.6%
	√	√	√	$A \overset{3/4\neg E}{\text{——}} B$	1,373	84.6%	80.3%
√		√	√	$A \overset{3/4\neg I}{\text{——}} B$	1,013	88.8%	82.2%
√				$A \overset{1/4E}{\text{——}} B$	34,155	100%	7.7%
	√			$A \overset{1/4I}{\text{——}} B$	13,997	100%	23.7%
√	√			$A \overset{1/2}{\text{——}} B$	114	90.4%	57.9%
√		√		$A \overset{1/2E}{\text{——}} B$	19	78.9%	47.4%
√			√	$A \overset{1/2AB}{\text{——}} B$	143,342	100%	58.4%
	√		√	$A \overset{1/2I}{\text{——}} B$	51	72.5%	66.7%
Total (including Self-Peerings)					236,663		
Self-Peerings					195		

Table 4.5: Classification of the peerings discovered in the IRR

4.5 Experimental Results and Comparison with Previous Work

In order to compare the peerings discovered with our techniques with those discovered with previous approaches we ran on the same data set we used in our experiments the piece of code used for peering extraction in the RRCC service [SGK+01, rrc]. The peerings obtained in such a way can be considered analogous to our peering candidates. By using the RIPE code we obtained 295,587 *RRCC peering candidates*, that are much less than our overall amount

of 512,758 peering candidates (see Tab. 4.4). By aggregating the RRCC peering candidates with the method of Section 4.4.3 we obtained 108,521 *RRCC peerings*. Again, much less than our 236,663 peerings (see Tab. 4.5). Further, there are 102 RRCC peerings that we did not find. We discovered that 100 of them involve private ASes and the remaining 2 come from an incorrect processing of the RRCC code of the `and` operator. A comparison with [CBD+02] gave similar results.

Comparing our results with the ones presented in [MKF+06,ZLMZ05,SF04] is not easy. In fact, they refer to the versions of the IRR of 04/07/04, 10/24/04, and 06/22/03, respectively. To the best of our knowledge, no repository is available with IRR historical data. We have a repository of such data in the interval described in Fig. 4.1 but, unfortunately, such interval does not cover the above dates. The authors of [MKF+06] provide the peerings extracted from the IRR on 04/07/04. The work in [ZLMZ05] is supported by a Web site providing several files of peerings. It is updated on a daily basis, yet the peerings discovered in the IRR are unavailable. Also the work in [SF04] has a Web site [nem] that allows to interactively explore the peerings detected on 11/08/05. Again, such date is not covered by our archives.

Hence, only a rough comparison is possible. The topology of [MKF+06] reports 56,973 peerings while [ZLMZ05] reports the discovery of 70,222 peerings. Both refer to the RIPE registry only. Paper [SF04] reports 127,498 peerings referred to the entire IRR. All such figures are very far from our results.

The above results, together with the data extracted by the on-line service over a period of two months, support the effectiveness of our peering extraction methodology compared to the state of the art.

4.6 Conclusions

In this chapter we describe how to extract peering relationships from the Internet Routing Registry. Namely, we first integrate information from different registries and solve possible inconsistencies, then we extract inter-AS peerings from RPSL objects, and finally we classify them according to their peering specifications. We also developed an on-line service which implements our methodology. Experimental results show that we discover many more peerings than the state-of-the-art techniques.

Chapter 5

Measuring Route Diversity from Remote Vantage Points

Recent works on modeling the Internet topology [MFM+06, MUF+07] have highlighted how the complexity of relationships between Autonomous Systems can not be oversimplified without sacrificing accuracy in capturing route selection. Such a shortcoming can mislead the understanding, hence the prediction, of the BGP behavior. In particular, models that assume an AS to be an atomic entity fail to account for route diversity, informally defined as the selection within a single AS of multiple routes to the same destination prefix.

Towards this goal, we devise a methodology to compute route diversity from a continuous stream of collected BGP messages. The analysis of our results shows that (*i*) accounting for the BGP dynamics allows to extract much more diversity than from a static snapshot of the Internet routing configuration; (*ii*) route diversity observed for an AS is strongly related to its location in the customer-provider hierarchy; (*iii*) the distribution of route diversity over ASes is unlikely to be biased by the specific choice of the collection system, while the number of prefixes exhibiting route diversity can depend on both number and location of the vantage points.

The main results presented in this chapter are also described in [DRCD09].

The rest of the chapter is organized as follows. In Section5.1, we describe previous work and summarize our contributions. Section 5.2 formalizes the metrics we use to model route diversity. In Section 5.4 we describe how we compute our metrics out of BGP update streams. Section 5.5 analyzes the results obtained by applying our approach to the dataset described in Section 5.3.

5.1 Introduction

The topological characteristics of the Internet have been the subject of several research efforts (e.g., [ZLMZ05, MFM+06]), which significantly helped understand the network behavior. In particular, the Internet is usually represented as a graph of Autonomous Systems, that exchange reachability information using BGP.

Previous Work

Recently, [MFM+06] proposed a model of interdomain routing which accounts for *route diversity*, i.e., the simultaneous usage in a single AS of multiple routes towards the same destination prefix. Basically, [MFM+06] relaxed the common constraint of ASes being regarded as atomic entities, by identifying in each AS multiple BGP routers which make different decisions about the routing of a specific destination. This model has been exploited to better predict the Internet behavior [MFM+06] and to study the interaction between the Internet topology and inter-AS policies [MUF+07]. However, the analysis performed in [MFM+06, MUF+07] only relied on a static snapshot of the network (i.e., a BGP's steady state), while several works (e.g., [DKR05, ZLMZ05]) underlined how routing dynamics provide a significant amount of additional information about the Internet topology.

Our Contributions

In this chapter we propose a technique that takes into account the BGP routing dynamics in order to extract route diversity from BGP data collected by passive remote vantage points. Using our approach, we identify much more diversity than in [MFM+06, MUF+07]. Since route diversity can be exploited to build BGP models that better fit the collected data, we believe that this improvement is relevant, because lots of research projects which aim at studying Internet dynamics rely on public data collected at remote vantage points.

We also provide a characterization of the route diversity as computed by our approach. First, we analyze the relationship between the route diversity observed for an AS and its location in the Internet customer-provider hierarchy. Moreover, we investigate whether the distribution of route diversity over all the ASes is biased by the specific choice of the dataset, in terms of number and location of observation points.

5.2 Modeling the Route Diversity in the Internet

We define the relationship $X_A \neq_{p,t} X_B$ to indicate that at time t AS X reaches the prefix p using two different routes, one through AS A, the other through AS B. Given that every BGP router can use only its current best path to p, the simultaneous usage of two distinct paths by AS X implies that X has at least two distinct BGP routers, one (X_A) peering with AS A and the other (X_B) peering with AS B. We formalize this intuition through the following property.

Property 1. *If there exist two distinct paths* $\pi_1 = (X, A, \dots)$ *and* $\pi_2 = (X, B, \dots)$ *that are both used by AS* X *to reach prefix* p *at time* t*, then* $X_A \neq_{p,t} X_B$*.*

Observe that paths π_1 and π_2 could share a common left subsequence of ASes. In such a scenario, we disambiguate the two (or more) routers that AS X uses to peer with as A by means of numerical superscripts (e.g., $X^1_A \neq_{p,t} X^2_A$).

The simultaneous usage by a single AS of multiple routes to the same prefix is commonly called *route diversity*. We distinguish route diversity from *path diversity*, where the latter refers to the simultaneous availability of multiple routes between the same AS pair.

We extend the definition of $\neq_{p,t}$ to a set of distinct routers $S_X = \{X_A, X_B, \dots\}$ (with $|S_X| \geq 2$), that we call *diversity set*. The relation $\neq_{p,t} S_X$ means that AS X uses a distinct router to peer with each one of the ASes $\{A, B, \dots\}$. More formally, $\neq_{p,t} S_X \Leftrightarrow \forall (X_A, X_B) \in (S_X \times S_X)$, $X_A \neq_{p,t} X_B$. We say that the diversity set S_X is *associated with* AS X.

As already pointed out in [MFM+06, MUF+07], $\neq_{p,t}$ relationships associated with AS X impose a lower bound on the number of BGP routers that X must employ to exhibit the observed route diversity. In particular, if we infer diversity sets by means of Property 1 and then search for the biggest diversity set associated with AS X across prefixes, we actually infer a lower bound on the number of BGP routers in X. We denote the cardinality of the biggest diversity set associated with X as its *max-rd*. More formally, a lower bound for the BGP routers of X is $\text{max-rd}(X) = \max_{S_X : \neq_{p,t} S_X} |S_X|$.

Consider the example in Figure. 5.1. AS 5 uses three distinct paths to reach AS 1, namely $\pi_1 = (5, 2, 1)$, $\pi_2 = (5, 3, 1)$, $\pi_3 = (5, 4, 1)$. Hence, by Property 1, the relation $\neq_{p,t} \{5_2, 5_3, 5_4\}$ holds and implies that AS 5 has at least 3 BGP routers. AS 6 uses two distinct paths to reach AS 1, both of which have AS 5 as a common neighbor. Hence, Property 1 implies that $6^1_5 \neq_{p,t} 6^2_5$.

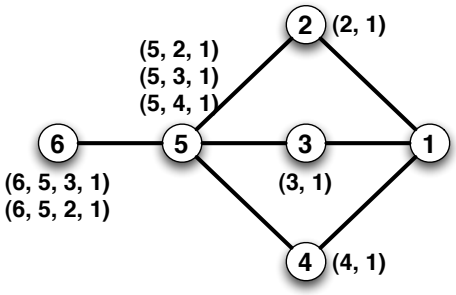


Figure 5.1: Example of network topology. Each node represents an AS and each edge one (or more) peering(s) between a pair of ASes. AS 1 origins prefix p. Each node is associated with a set of paths it uses to reach p at time t.

5.3 Our Dataset

The analyses shown throughout this chapter rely on BGP data (both BGP RIBs and updates) collected by 574 CPs of the RIS [roub] and RV [roua] projects from 07/17/2007 to 07/30/2007 (called hereafter *reference period*).

Our dataset contains 538,341,994 updates (about 38M updates per day on average) with 12,351,221 distinct AS-paths on 76,014 distinct peerings and 26,491 distinct ASes. The number of prefixes is 293,154.

For comparison, the dataset used in [MFM+06,MUF+07] consists of a single snapshot of RIBs collected on 11/13/2005 from more than 1,300 observation points and contains 4,730,222 distinct AS-paths on 58,903 distinct peerings. Note that, even if the number of observation points is more than twice than ours, our dataset contains many more peerings and AS-paths. Hence, since route diversity can be inferred by comparing distinct paths collected at different vantage points (see Property 1), more AS-paths can result in the inference of more route diversity. We highlight that, while the two datasets refer to different time periods, we ascribe the difference in terms of number of distinct AS-paths mostly to the contribution of BGP updates. This intuition is backed by [DKR05, ZLMZ05], which show that taking BGP dynamics into account systematically results in more topology information than relying only on a static snapshot of the Internet.

5.4 Extracting Diversity Relationships in a Dynamic Setting

How to extract diversity relationships from a static snapshot of the network has been already discussed in [MFM+06]. We move one step further by dealing with the problem of extracting $\neq_{p,t}$ relationships in a dynamic setting, taking into account routing changes. One of our goals is to evaluate how BGP dynamics affects the number of diversity relationships that can be inferred from data collected by multiple remote vantage points.

Our approach is as follows. We scan in chronological order the stream of BGP updates collected in a given time interval by all the CPs and we keep track of which paths are used by which AS over time. Namely, for every AS X and prefix p we keep in memory the *used paths set* $ups(X, p, t)$, i.e., the set of paths that are used by X to reach prefix p at time t, as observed by at least one CP. We also associate to every $ups(X, p, t)$ the number of CPs which observe prefix p through AS X at time t. By using a reference-counting like mechanism, we update *ups* by removing a path π as soon as no CP sees AS X reaching p through π anymore. Finally, we infer $\neq_{p,t}$ relationships associated with AS X by applying Property 1 to the paths in $ups(X, p, t)$.

Our implementation computes diversity sets out of a stream of BGP updates in near real-time. Namely, it took us less than 12 hours to run this computation on our whole reference period, using on average 2.4 GB memory on a common machine (2x Intel Xeon 2.80GHz CPUs, 4GB RAM).

It is important to note that BGP prefix aggregation can mislead our approach. In fact, the AS-path carried in an aggregated announcement does not identify a unique route to the destination prefix [RLH06]. Thus, we deliberately disregard such announcements. We argue that this is a negligible limitation, since BGP prefix aggregation is renowned to be seldom performed. E.g., in our dataset, a RIB contains on average less than 100 aggregated announcements ($< 0.05\%$).

When extracting diversity sets from BGP updates, a major concern is handling transient periods, because it could produce false positives. Consider two collector peers cp_1 and cp_2, both having a route $(X\ Y\ Z)$ to prefix p at time t_0. Assume that the peering between X and Y fails at time t_1 ($t_1 > t_0$), and cp_1 switches to an alternative path, say $(X\ W\ Z)$ at time t_2 ($t_2 > t_1$). Due to the propagation delay of BGP messages, cp_2 could still be using path $(X\ Y\ Z)$ at time t_2. As a consequence, Property 1 applies; hence our algorithm would incorrectly infer $X_Y \neq_{p,t} X_W$. Observe that BGP *path exploration* - i.e. dis-

carding alternative paths that are gradually withdrawn (see, e.g., [OZP+06]) - additionally exacerbates this problem. We face such a shortcoming by defining a *convergence time* Δ. Namely, when computing the used paths set, any prefix that underwent a routing change in the last Δ seconds is not considered for the inference of diversity relationships.

Different choices of Δ can lead to different diversity sets, potentially affecting the value of max-rd for a number of ASes. Namely, choosing a Δ smaller than the time it takes to update all the collector peers about a given event might result in false-positives. On the other hand, using bigger values of Δ, we may disregard paths which actually represent diverse routes, producing false-negatives.

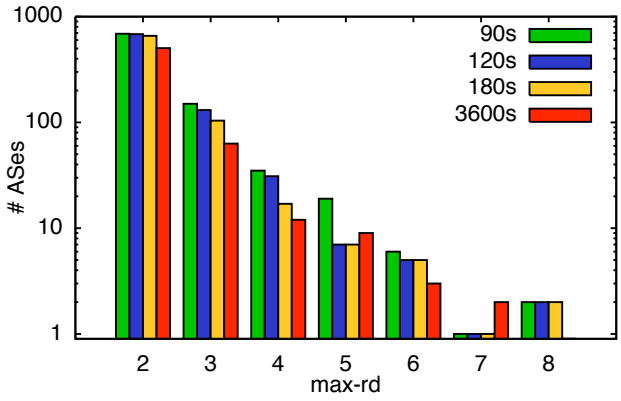


Figure 5.2: max-rd computed over the reference period using different values of the convergence time Δ. ASes with max-rd = 1 are omitted.

In order to assess this sensitivity, we ran our algorithm on our dataset with different values of Δ. Fig. 5.2 shows the distribution of max-rd for $\Delta = 90$, 120, and 180 seconds. The distributions are almost unaffected by the specific choice of Δ. As a further proof, we plot the distribution obtained by setting $\Delta =$ 3600 seconds, which is an extremely conservative convergence time. Observe that the values of max-rd are still quite comparable with the ones obtained with realistic values of Δ. Backed from the results in [LWVA01], throughout this chapter we will assume $\Delta = 180$ seconds.

5.5 Understanding Route Diversity in the Internet

In this Section we will characterize the route diversity phenomenon with respect to the metrics defined in Section 5.2 applied to our dataset.

5.5.1 The Impact of BGP Dynamics on Route Diversity

As discussed in Section 5.4, our methodology extends [MFM+06] by accounting for the BGP dynamics. We will show the effect of extracting route diversity from BGP updates.

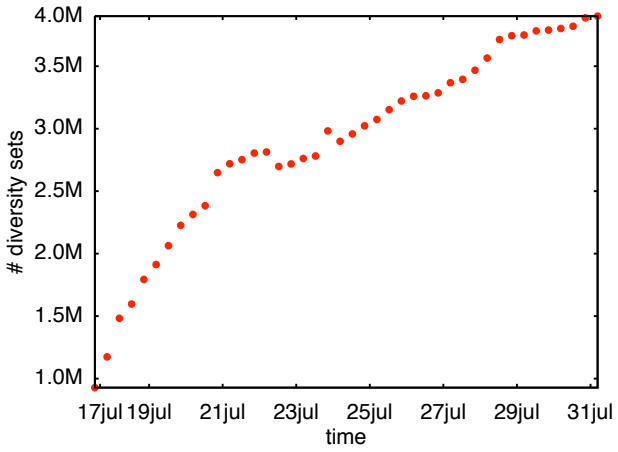


Figure 5.3: Cumulative number of diversity sets over time.

Figure 5.3 plots the cumulative number of distinct diversity sets inferred over time. In particular, the first data point represents the number of diversity sets extracted from the initial RIBs, while the data point at time t represents the number of diversity sets extracted using both the initial RIBs and the BGP updates collected up to t. This number significantly increases over time. In particular, at the end of the reference period, we infer approximately four times the diversity sets obtained from the initial RIB snapshot. Note that the growth exhibits a pseudo-logarithmic trend, i.e., taking BGP dynamics into account can greatly influence the ability to capture route diversity only on relatively short time scales. On the other hand, considering larger time

scales can potentially mislead the inference, because of changes in the network topology

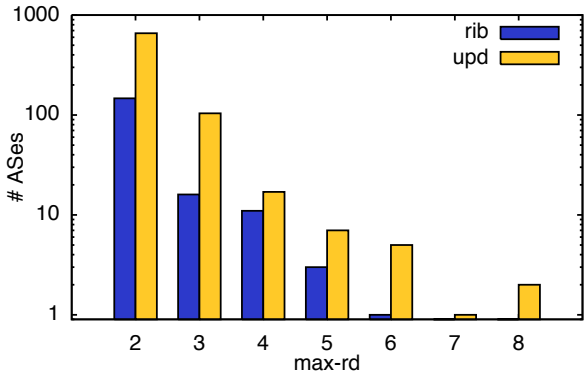


Figure 5.4: max-rd extracted from a RIB snapshot at the beginning of the reference period and from all the BGP updates collected during the reference period. ASes with max-rd = 1 are omitted.

Moreover, accounting for BGP dynamics allows us to identify a higher number of ASes exhibiting max-rd > 1. Namely, we detected 795 such ASes over the whole reference period, while only 178 from the initial snapshot. Breaking down such ASes according to their values of max-rd, Figure 5.4 shows a significant improvement in the distribution of max-rd at the end of the reference period, compared to the values computed on the initial RIBs.

Overall, our results show that BGP dynamics play an important role in detecting the diversity relationships that can be observed by remote vantage points, hence our methodology can help build topology models where route diversity is captured more accurately.

5.5.2 Sensitivity to Our Dataset

In order to assess the scope of our results, we aim at understanding up to what extent the diversity relationships we extracted depend on our dataset.

Since max-rd relies on the availability of distinct paths as observed by any CPs, we analyze whether its value can be biased towards those ASes that host a large number of collector peers. Figure 5.5 plots the relationship between

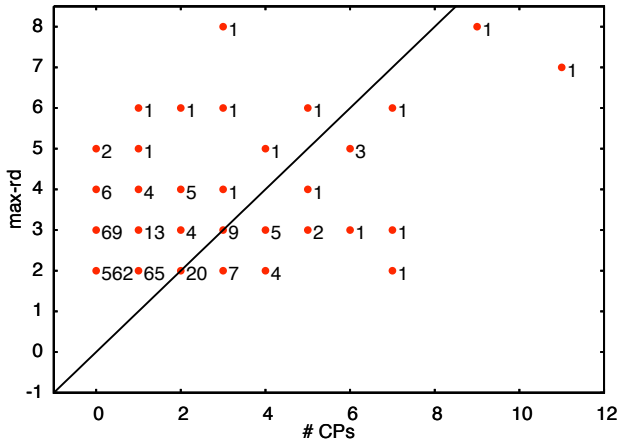


Figure 5.5: Each point (x, y) is labelled with the number of ASes that host x CPs and have max-rd $= y$. $y = x$ line is also plotted for convenience.

number of CPs hosted by an AS and its max-rd. It shows that most of the diversity relationships are inferred for ASes hosting a few (or no) CPs, while a small number of ASes shows very low values of max-rd, although they host several CPs.

Moreover, the presence of a CP could influence not only the max-rd of the AS that hosts it, but also of others. To verify if this happens, we randomly partitioned the whole set of CPs into four sub-sets and we computed the max-rd over each of them. Figure 5.6 shows that max-rd distributions of all the sets are quite similar and closely resemble the max-rd distribution over all the CPs.

Overall, the max-rd metric seems to be only minimally affected by the number and the location of collector peers. As further comparison, we also note that the max-rd distribution at the beginning of the reference period (Figure 5.4) closely follows what presented in [MFM+06], both in trend and in values, even if the data sources greatly differ both in the time period and in the number and location of the used vantage points.

Unfortunately, this property does not hold for the number of prefixes exhibiting route diversity. (Note that, overall, more than 230k prefixes in our dataset exhibit some route diversity - about 78% of all routed prefixes.) Figure 5.7 plots the relationship between the ASes hosting no CP and exhibiting

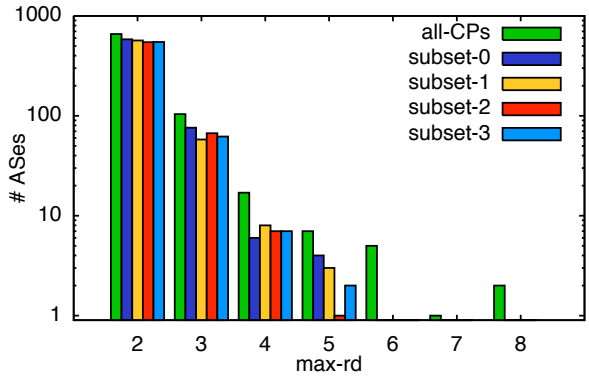


Figure 5.6: max-rd extracted from subsets of CPs. ASes with max-rd = 1 are omitted.

some route diversity and the prefixes the diversity is observed on. Note that the distribution clearly exhibits a power-law trend. By contrast, the relationship between the ASes hosting CPs and exhibiting some route diversity and the prefixes the diversity is observed on (Figure 5.8) features an irregular distribution. Hence, we suspect that the number of prefixes for which some route diversity can be observed in a given AS is a metric that can be heavily biased by the choice of vantage points.

5.5.3 Relating Route Diversity to the Internet Hierarchy

We now investigate which types of ASes exhibit route diversity that can be observed by remote vantage points. Our intuition is that ASes with a large number of customers most likely maintain multiple routes toward the same destination, in order to ensure reliable connectivity. To estimate the number of customers, we relied on data provided by CAIDA [cai]. Namely, CAIDA publishes a ranking of the ASes based on the cardinality of their customer cones, estimated as described in [DKF+07]. Basically, the topmost ASes in such a ranking are supposed to have the largest number of customers in the Internet.

We thus compared the max-rd with CAIDA's ranking related to the same reference period. Figure 5.9 shows that ASes with high values of max-rd usually have lots of customers and, on the other hand, ASes with low values of max-rd

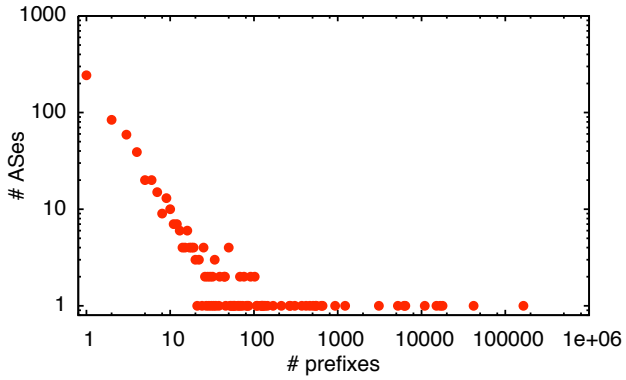


Figure 5.7: Each point (x, y) represents y ASes that host no CP and exhibit max-rd > 1 on x prefixes.

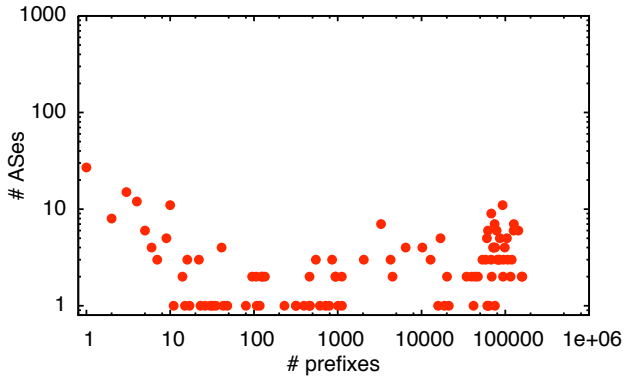


Figure 5.8: Each point (x, y) represents y ASes that host at least one CP and exhibit max-rd > 1 on x prefixes.

generally have a few customers. Figure 5.10 further stresses the difference of max-rd distribution of ASes belonging to different portions of the ranking (namely, top-n ASes, with $n \in \{10, 50, 100\}$).

Our analysis shows that the location of ASes in the Internet hierarchy strongly impacts the diversity that can be observed by remote vantage points.

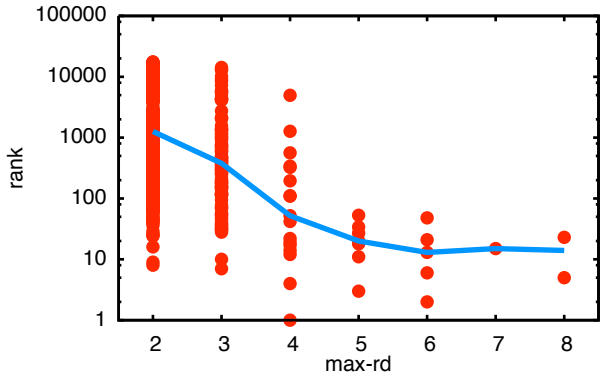


Figure 5.9: Each point (x, y) represents an AS having max-rd $= x$ and CAIDA's rank $= y$. The solid line shows the median value of rank among ASes having the same value of max-rd.

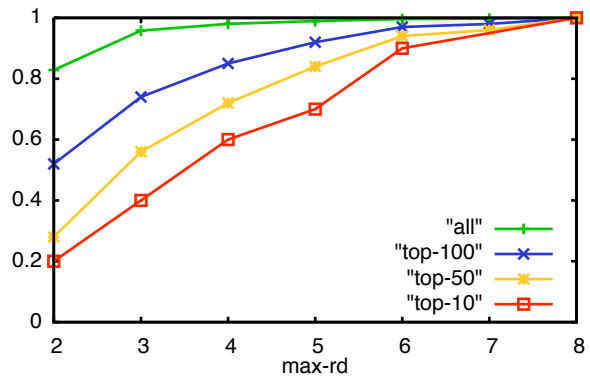


Figure 5.10: CDF of max-rd for the top n ASes, $n \in \{10, 50, 100\}$. *all* refers to the whole topology.

5.6 Conclusions

In this chapter we studied the characteristics of the BGP route diversity that can be observed by passive remote vantage points. We defined a methodology to extract diversity relationships over time from a continuous stream of BGP updates. By accounting for the routing dynamics, we inferred four times more

diversity relationships than only using a static snapshot of the Internet routing configuration. We also showed that the behavior of the max-rd metric is unlikely to be biased from the specific choice of the dataset. Finally, we correlated the observed route diversity of ASes to their position in the Internet hierarchy.

Part II

Case Studies

Chapter 6

Mediterranean Fiber Cut

On the morning of 30 January 2008, two submarine cables in the Mediterranean Sea were damaged near Alexandria, Egypt. The media reported significant disruptions of Internet and phone traffic in the Middle East and South Asia. About two days later, a third cable was cut, this time in the Persian Gulf, 56 kilometers off the coast of Dubai. In the days that followed, more news on other cable outages came in.

We looked at the impact these events had on Internet connectivity by analyzing the data collected by the *Routing Information Service* (*RIS*) [roub] of *RIPE NCC* [ripa] and using publicly available tools developed by the *Compunet Research Group* of *Rome Tre University* [com] and by the RIS.

The main results presented in this chapter are also described in [ACK+08] and it has also been published as a RIPE NCC's document at [ripd].

6.1 Background

The history of submarine telecommunications cables goes back to 1850, when the first international telegraph link between England and France was established. Eight years later, the first trans-Atlantic telegraph cable linked Europe to North America. In the 20th century, telephony became the driving force for submarine cable deployments. TAT-1, the first trans-Atlantic telephone cable, was installed in 1956. It had the capacity to transmit 36 analog phone channels simultaneously. These days fibre-optic submarine cables carry the bulk of the trans-oceanic voice and data traffic. Maximum capacity is now in the order of 1 Tb/s, equivalent to 15 million old analog phone channels. Compared to satellites, submarine cables offer higher capacity and, because of the shorter distance, feature much better latencies. Cable systems are also more cost effective on major routes. In the past two decades, many of these cables have been deployed, primarily triggered by the explosive growth of Internet traffic.

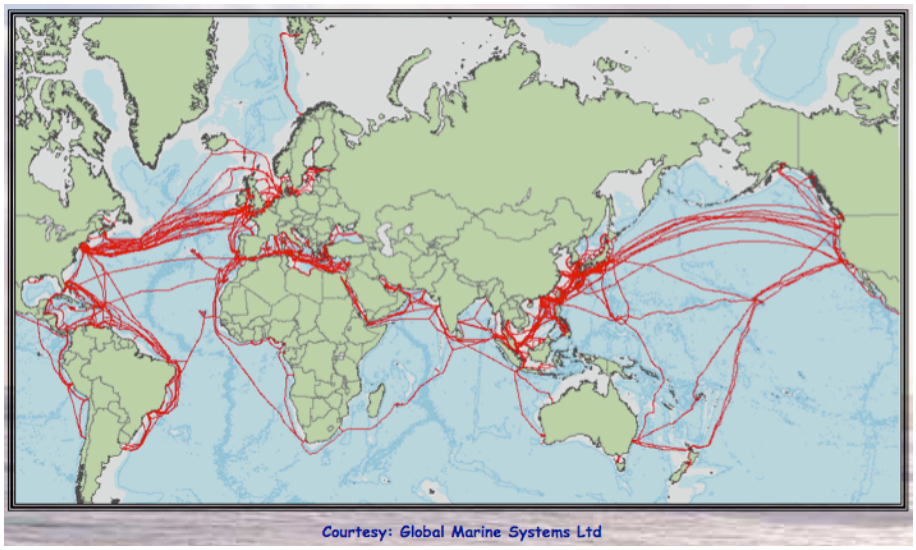


Figure 6.1: World map of submarine cable systems.

The world map of cable routes (Figure 6.1) shows that Europe, North America and East Asia are well connected. Numerous cables connect continents and

countries. However, Africa, the Middle East and South Asia have far fewer cable systems. Looking at the available bandwidth or capacity in these cables, the differences become even more apparent. Faults in cables connecting these regions therefore have a higher impact than comparable faults in trans-Atlantic cables.

Although they rarely make news headlines, cable faults are not uncommon. Global Marine Systems, a company active in submarine cable installations and repairs, reported more than 50 failures in the Atlantic alone in 2007. A study [medb] on behalf of the Submarine Cable Improvement Group shows 75% of all faults are caused by external aggression (physical damage). Of these, three out of four were attributable to human activities such as fishing, anchors and dredging. Natural hazards such submarine earthquakes, density currents and extreme weather were responsible for the remainder.

To limit the impact of such faults, cable systems often have build-in redundancy. Ring structures, for example, cross the ocean twice, each cable segment taking a geographically different route. When one segment breaks, signals can still reach a destination over the other segment(s). Repairs can then take place without much media attention.

6.1.1 Location of the Mediterranean Cables

The International Cable Protection Committee provides lists of deployed cable systems on their website [meda]. Several smaller cables connect most regions bordering the Mediterranean. Spain, Morocco, France, Algeria, Italy, Tunisia, Greece, Libya, Cyprus, and Israel have multiple connections. However, Figure 6.2 shows that only three cables connect Europe to Egypt, the Middle East and Asia: Flag Europe Asia, SEA-ME-WE4 and its predecessor SEA-ME-WE3. When the first two failed on 30 January 2008, the low capacity SEA-ME-WE3 was left as the only cable system providing a direct route from Europe to the region. All other options for rerouting traffic involved much longer cable routes or satellite systems.

6.1.2 Effects of a Cable Cut

When a communications cable system used for IP connectivity fails, two things can happen:

- Networks become unreachable, meaning they disappear from the Internet; or

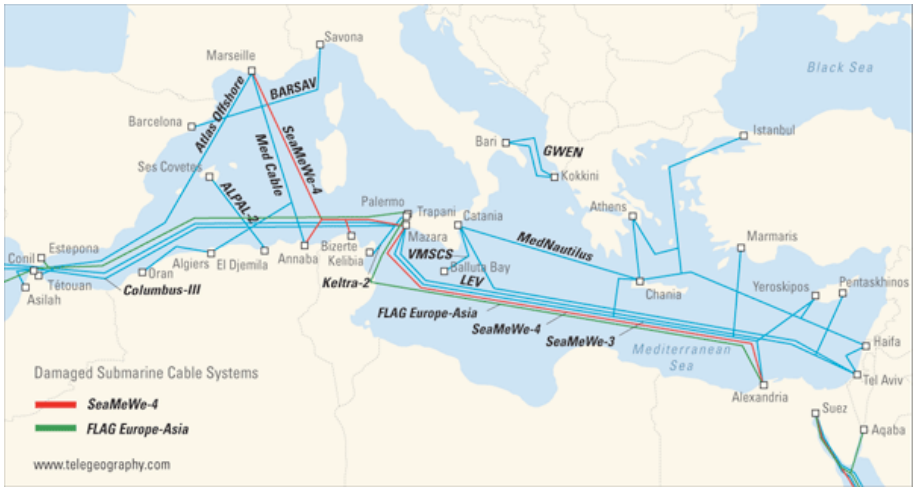


Figure 6.2: Mediterranean cables.

- Traffic is rerouted.

Every individual IP link set up over the failed cable will be subject to one of these two options. Both options, however, can refer to a number of specific scenarios:

- Networks may become unreachable, because no action is taken, meaning packets fall in a "black hole"
- Networks may become unreachable, because the (only) upstream provider withdraws route announcements.
- Traffic may be rerouted on the IP level, either by manual reconfiguration or by routing protocols like BGP reacting to a loss of IP connectivity to previously preferred routers.
- Traffic may be rerouted on the data link layer. The (virtual) circuits on which an IP link has been set up are changed to follow a different physical path.

In the analysis below, we see evidence that all of the above occurred after the outages on the two Mediterranean cables. The outages on the FALCON cable are not visible in our data. If the FALCON cable failures had significant effects on Internet connectivity, these were obscured in our data by the network outages, the network congestion and the rerouting activities triggered by the problems in the Mediterranean.

6.2 Event Locations

Figure 6.3 shows where the events were located.

Figure 6.3: Google map of the cable cuts.

6.3 Event Timeline

The following events are known/confirmed:

Wednesday, 23 January 2008 (exact time unknown) FALCON cable, segment 7b damaged (Persian Gulf) Note: This is one week prior to the Mediterranean outages.

Wednesday, 30 January 2008, 04:30 (UTC) SEA-ME-WE-4 cable, segment 4/Alexandria-Marseilles, 25 kilometers from Alexandria, Egypt.

Wednesday, 30 January 2008, 08:00 (UTC) FLAG Europe-Asia cable (FEA), segment D (EG-IT) cut approximately 8.3 kilometers from Alexandria, Egypt.

Friday, 1 February 2008, 05:59 (UTC) FALCON cable, segments 2 and 7a (AE-OM) cut approximately 56 kilometers from Dubai, UAE.

Friday, 1 February 2008 (exact time unknown) Unidentified cable, between Halul (QA) and Das (UAE).

Friday, 8 February 2008 (exact time unknown) SEA-ME-WE-4 repair completed.

Saturday, 9 February 2008, 18:00 (UTC) FEA segment D repair completed.

Sunday, 10 February 2008, 10:00 (UTC) FALCON cable repair completed.

Thursday, 14 February 2008 Doha-Halul part of the unidentified QA-UAE cable "to be operational soon".

Note: Date and time for FEA segment D and FALCON segments 2 and 7a cable outages were reported by FLAG Telecom on their website. Other dates are from news reports. The SEA-ME-WE4 operators have not published exact times of cable failure and repairs. The timestamps for SEA-ME-WE4 failure are from clear observations in measurement data (both BGP monitoring and active measurements).

6.4 Dataset

Our analysis focused on publicly available data from the following RIPE NCC [ripa] services:

***Routing Information Service* (*RIS*)) [roub]** The RIS collects Border Gateway Protocol routing information messages from 600 peers with 16 collection boxes called Remote Route Collectors in near real time. Results are stored in a database for further processing by tools such as BGPlay [bgpb], a visualization tool. Three times a day, the route collectors take snapshots of their respective Routing Information Bases (often referred to as *RIB dumps*). Figure 6.4 shows the location of the RIS Remote Route Collectors.

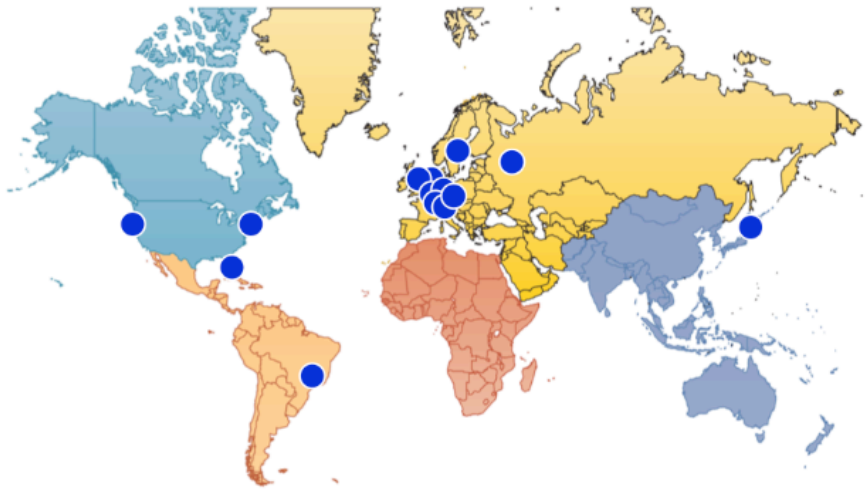


Figure 6.4: Location of RIS Remote Route Collectors.

Test Traffic Measurements (TTM) TTM measures key parameters of connectivity between a site and other points on the Internet. Traceroute vectors, one-way delay and packet-loss are measured using dedicated measurement devices called *test-boxes*. Configured in an almost full mesh (that is, fully inter-connected), the TTM test-boxes continuously monitor end-to-end connectivity between the hosting sites. Figure 6.5 shows the location of the TTM nodes.

DNS Monitoring Service (DNSMON) DNSMON provides a comprehensive, up-to-date overview of the service quality of root name servers, as well as participating country codes and generic Top Level Domains (TLDs). Using the infrastructure of TTM probes, DNSMON measures query response times to 196 name servers. Therefore DNSMON provides another perspective on possible effects the cable outages may have had on the Internet.

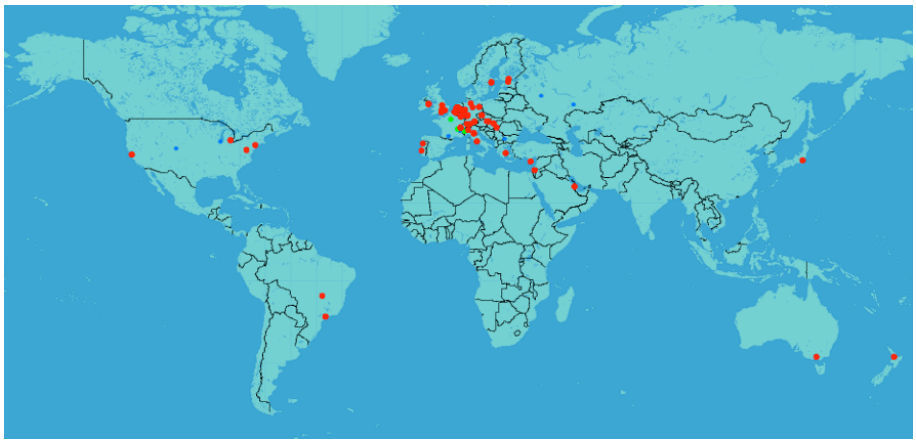


Figure 6.5: Location of TTM nodes.

6.5 Analysis

In this section, we discuss the results of each of the services described above. However, we want to stress that the bigger picture only emerges when we combine the knowledge gained in the separate fields. Where BGP looks at routing information, active measurements by TTM and DNSMON provide a glimpse on how networks performed during the outage. Results from one service thus help us understand results from the others.

6.5.1 BGP Overall

The impact of the outages on BGP largely depended on how the cables were used in peering sessions.

- When peering between a Middle East-based provider and their transit in the West was established over the cables, or when the peering depended on receiving reachability information over the cable through some interior routing protocol, the sessions were dropped immediately after the failure. If the provider had no other way to route its IP traffic, the networks disappeared from the global routing tables at 04:30 or 08:00 (UTC) respectively.

- In cases where peering happens in the West, or where Middle Eastern networks are announced statically in the West, the sessions would not be expected to go down immediately after a cable failure. Instead, visibility in BGP would depend on manual network administrator action. This explains why some networks disappeared from the global routing tables several hours after the cable failures.

Where backup paths did exist, BGP explored and used them. We observed significant changes in the peering usage between larger transit providers. However, due to limited bandwidth and increased demand on the backup paths, sessions weren't always stable. In those cases BGP had difficulties converging on alternate topologies.

6.5.1.1 Prefix Counts

The number of prefixes seen in RIS is an indication of global network reachability. If a prefix cannot be seen and is not covered by an aggregate, it is likely the network in question has become unreachable. Therefore, total prefix count is a good starting point for BGP analysis. Using the data from the eight hourly RIB dumps, we graphed the total number of prefixes seen by RIS over time (Figure 6.6). The result does not show any clear drop or increase in the amount of prefixes near the known failure and recovery events. On a global scale, the cable outages only affected a small percentage of all networks.

However, when we assign country codes to each prefix, based on the information from RIR delegation statistics [ripe], we see various countries with significant reductions in the number of announced prefixes. Figure 6.7 compares, for each country, the number of prefixes seen in RIS over time with the number of prefixes visible at 00:00 (UTC), 30 January 2008. This gives an indication of the impact of the cable failures on network reachability in BGP. Egypt, Sudan and Kuwait were amongst the hardest hit, with drops of up to 40% in prefix visibility.

6.5.1.2 Analysis of AS Path Changes

Next we looked at changes in Autonomous System (AS) paths. In relation to the cable cuts, the two main reasons for change are:

- Networks disappearing from the routing tables (this reduces the total number of AS paths)

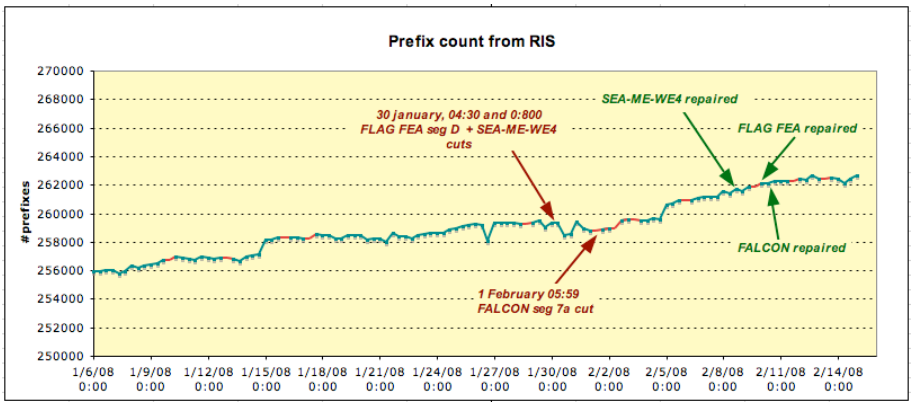


Figure 6.6: Total prefix count vs. time.

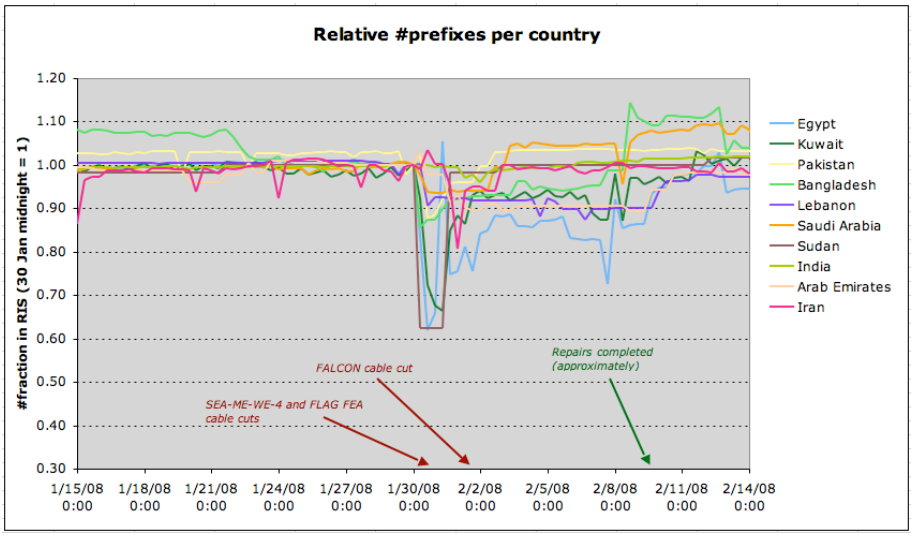


Figure 6.7: Relative prefix count for most affected countries.

- Networks rerouted on IP level (when the preferred transit is unavailable, BGP will try backup paths; these are likely to be longer)

Again starting from the RIB dumps, we compared consecutive dumps from the largest RIS route collector, RRC03, and determined the changes in distinct AS paths. Because AS path changes capture more of the event, we thought that graphing global level changes would clearly and unambiguously show when the cables snapped. That hypothesis did not hold. Although the cable cuts triggered a flux of changes, it is not the only such signal in a one month time period. Other events, either in BGP or related to RIS collector peers, triggered comparable levels of change.

As with the prefix counts, the correlation with the cable outages only becomes clear when we restrict the comparison to those AS paths where one or more of the constituent AS Numbers is registered to the region by an RIR.

Figure 6.8 shows the total number of distinct AS paths and the relative amount of change in distinct AS paths associated with the region. The changes seen align quite well with the event timeline.

Figure 6.9 again shows the total number of distinct AS paths associated with the region, this time augmented with the average length of those AS paths. As expected, the average AS path length increases around the time period of the cuts.

6.5.1.3 Affected BGP Peerings

During the fibre cuts, we observed significant changes in BGP peerings usage. Some peerings went down and were not used again before the repair of the fibre(s). Others were selected as backup paths, which meant that their usage exploded. The metric we used to approximate the usage of a BGP peering between two Autonomous Systems is the size of the sets of all the prefixes routed through the peering, as seen by all the available collectors at once.

From our data sources, we extracted the BGP peerings which, from 16:00 (UTC), 29 January to 08:00 (UTC), 3 February, matched the following criteria:

- Carried, at some time, a significant number of prefixes (that is, more than 50 distinct prefixes) as observed from all the available collector peers
- Routed no traffic (that is, zero prefixes world-wide) for more than 12 hours

590 distinct BGP peerings involving 309 distinct Autonomous Systems matched the criteria described above. The ratio between the number of peer-

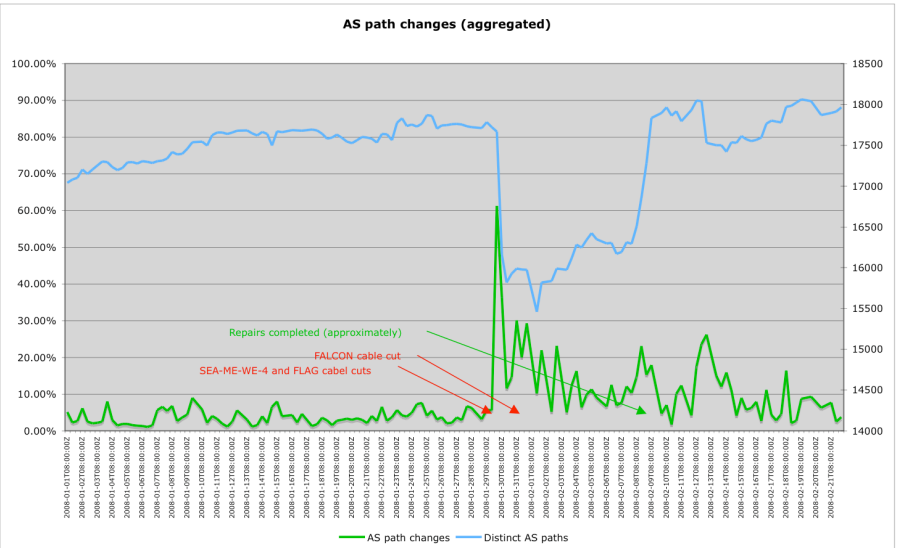


Figure 6.8: AS path changes (Aggregated).

ings and the number of Autonomous Systems suggests a high correlation of events.

As expected, AS15412 (FLAG Telecom) was one of the most involved Autonomous Systems, being present in more than 40 peerings. Other highly affected Autonomous Systems included AS4788 (TMNET, Malaysia), AS7575 (AARNET, Australia) and AS7473 (Singapore Telecommunications Limited), which were involved in more than 70 peerings overall.

The most affected AS was AS8966 (Emirates Telecommunications Corporation), which accounted for more than 100 peerings. From communications on the LINX-ops mailing list, we learned that AS8966 stopped advertising prefixes at the London Internet Exchange because of lack of capacity on back haul links. This meant that all peers at LINX explored alternative paths towards AS8966.

Note that the selection criteria for the peerings will give us the failing links, links which went down on purpose and links which were used for backup connectivity.

6.5. Analysis

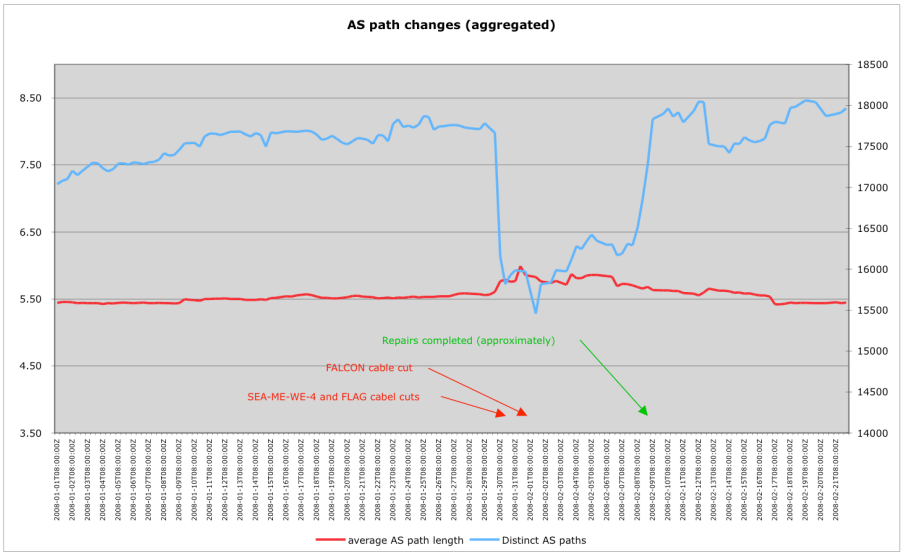


Figure 6.9: AS path changes (Aggregated).

Backup links As shown in Figures 6.10 and 6.11, some previously unused peerings suddenly began to route a large number of prefixes, probably due to the unavailability of other (preferred) routes.

Large sets of prefixes were rerouted and subsequently withdrawn within a few hours, thus BGP convergence likely slowed down.

Failing links As expected, some of the peerings experienced a drop in the number of prefixes routed right after the faults.

6.5.1.4 Analysis of BGP dynamics (Case Studies)

To better understand the routing dynamics caused by the fibre cuts, we analyzed some specific cases in detail, as samples of different patterns in routing changes.

The fibre cuts meant that some prefixes were unreachable for a significant period of time. Section 7.3 shows that the link between AS20484 (Yalla Online,

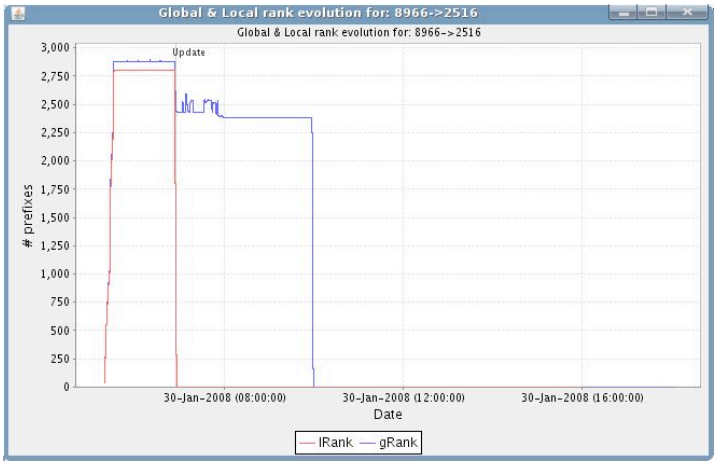


Figure 6.10: More than 25 hundreds prefixes were suddenly routed through the peering AS8966-AS2516 (Emirates Telecommunications Corporation - KDDI). This continued for a couple of hours.

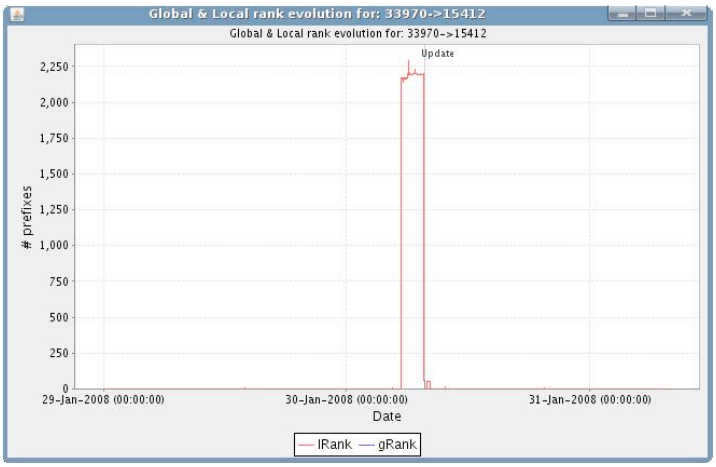


Figure 6.11: AS33970 (OpenHosting) temporarily routed a large set of prefixes through FLAG Telecom (AS15412). It took more than two hours to return to the previous usage level.

6.5. Analysis

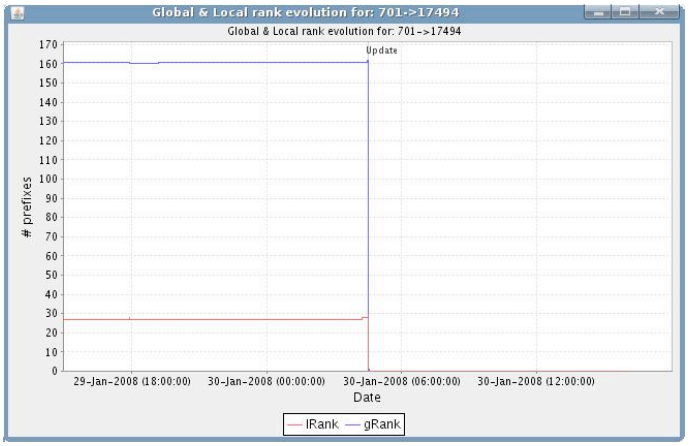


Figure 6.12: The peering between AS701 (UUNet) and AS17494 (Bangladesh Telegraph and Telephone Board) dropped by over 100 prefixes.

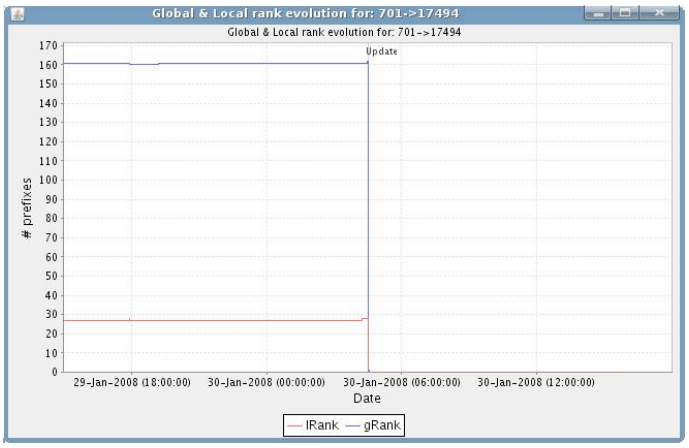


Figure 6.13: The AS701-AS17494 peering again, observed from a different collector peer. The local rank differs because the two peers see different numbers of prefixes with AS701 in their AS path. Obviously, the global rank evolution is the same as in Figure 6.12 (but note the slightly different time range on the x-axis).

Egypt) and AS8452 (TEDATA, Egypt) experienced a major network event; the prefixes usually routed through this link either were unreachable or changed

their routes for most of the period.

On the other hand, some prefixes did not change their routes. Section 7.3 demonstrates that the network hosting TTM box 138 (in Bahrain) was rerouted at sub-IP level. Thus its BGP routing didn't change, while the traffic experienced some major delays and packet loss; evidently, the backup bandwidth is much less than the original path over SEA-ME-WE4.

Other prefixes changed their routes for a significant period of time. Section 7.3 illustrates how AS17641 (Infotech, Bangladesh) changed one of its upstream providers.

Finally, we observed prefixes which experienced long periods of BGP instability. Section 7.3 shows how OmanTel triggered over 10,000 BGP update messages in RIS in a 90 hour (3.5 day) time period. Routes were flapping constantly for several peers, making it questionable whether BGP path exploration always converged.

Further details on each of these case studies can be found in the appendix.

6.5.2 Active Measurements

6.5.2.1 Test Traffic

Looking at statistics and plots published on the Test Traffic Measurement (TTM) website, we learned that from the approximately 75 active probes, only one had serious trouble as a result of the cable outages: node TT138, installed with 2connect in Bahrain, became unreachable from all other nodes at 04:30 (UTC) on 30 January.

After two and a half days, basic Internet connectivity was restored. However, the latencies and packet loss, especially for traffic going to Bahrain, were much higher. With peaks in one-way delay of 1.2 full seconds on 6 and 7 February, the network's performance would have been rather poor for end users. Finally, in the evening of 8 February, when repairs to the SEA-ME-WE4 cable had been completed, latencies returned to normal conditions.

Figures 6.14 and 6.15 show packet delay (black dots) and number of hops in traceroute (red lines and dots) to and from the RIPE NCC over time. Similar patterns were seen in plots to and from other TTM probes.

As illustrated in Section 7.3, the BGP routing information was relatively stable during this entire period. Even after the site became unreachable, BGP continued propagating routing information for 2Connect's prefix. We see this reflected in the traceroutes conducted by TTM: right after the cable cut, traceroutes from TT01 did start tracing the route to TT138, but replies stopped

6.5. Analysis

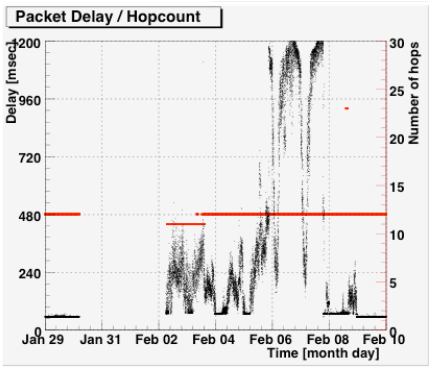


Figure 6.14: Delay from tt01 to tt138.

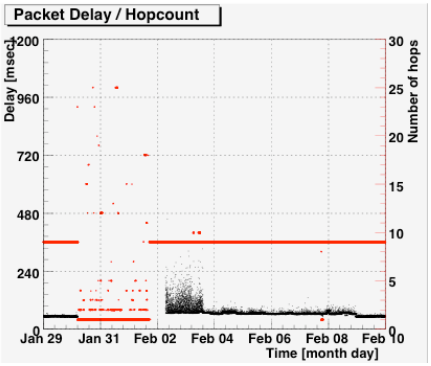


Figure 6.15: Delay from tt138 to tt01.

coming in after the eighth hop, in AS6453.

Traceroute results from Amsterdam to Bahrain on January 30th:

Before the cut, traceroute entered the destination AS at hop 9. The return-trip times (RTT) returned in manual traceroutes to TT138 suggest this hop is located in Western Europe. Interestingly, a traceroute to the hop 9 address, 80.88.244.33, returns with last hop IP 195.219.189.62. Because this address is from a network registered in RIPE Database as LONDON-TGB/Teleglobe's backbone, we conclude this network node is a Teleglobe router in London configured with IP addresses from Bahrain-based 2Connect. Teleglobe are also

hop	IP	origin AS
1	193.0.0.238	3333
2	195.69.144.110	1200/31283/30132/12989
3	4.68.120.15	3356
4	4.68.110.226	3356
5	80.231.80.5	6453
6	80.231.80.30	6453
7	195.219.195.6	6453
8	195.219.189.106	6453
9	80.88.244.33	35313
10	no response	
11	80.88.240.121	35313
12	80.88.240.14	35313

Table 6.1: Before the cable cut.

hop	IP	origin AS
1	193.0.0.238	3333
2	195.69.144.110	1200/31283/30132/12989
3	4.68.120.15	3356
4	4.68.110.226	3356
5	80.231.80.5	6453
6	80.231.80.30	6453
7	195.219.195.6	6453
8	195.219.189.106	6453
9	no response	
10	no response	
11	no response	
12	no response	
...		
30	no response	

Table 6.2: After the cable cut.

likely to be the ones injecting the 80.88.240.0/20 prefix into BGP on behalf of AS35313. This is done irrespective of the state of the link(s) to Bahrain, presumably using statically configured routes.

When basic end-to-end connectivity had been restored on 2 February, traceroutes to Bahrain showed a slightly different path in the final three hops. However, traceroutes from Bahrain don't show any change. We conclude that there was a temporary recovery from the SEA-ME-WE4 cut, created by Teleglobe setting up a different data link for IP traffic to Bahrain, and possibly augmented by a minor internal IP level change in Bahrain.

6.5.2.2 DNSMON

Because the DNSMON service uses the TTM infrastructure, problems with the Bahrain node are also showed in Figure 6.16.

The plot shows that none of the 196 monitored name servers answered queries from TT138 in the period for which TTM already reported problems. This is further evidence that, despite the routing information carried in BGP, the ISP was actually cut off from most of the Internet for two and a half days.

At first glance the list of monitored name servers and TLDs in the graph suggests DNSMON had no measurement targets in the Middle East or South Asia. However, this is not completely true. Some of the name servers are implemented with anycast, with a number of servers connected at different locations worldwide, all using the same global IP address. Which of these is actually used by a client depends on BGP policies, which affect how the name server's network is announced and received at various places.

The K-root name server, operated by RIPE NCC, is one example of an anycast server. Three instances are deployed in the region affected by the cable cuts: two of these, with Qtel in Doha (Qatar) and with Emix in Abu Dhabi (United Arab Emirates), are intended as local nodes; routing announcements for these K-root sites should not be propagated by the local peers to the global Internet. The third one in Delhi, India, is a global node; this means the peers can pass the announcements to their upstreams, thereby making the node an option to choose from for anyone who receives the route to Delhi.

In normal reporting DNSMON does not show which anycast instance was used by the probes. However, the raw data also includes queries into the identity of the server. Using this raw data, we looked for correlations between k.root-servers.org instance changes on each DNSMON probe and the known cable outage events. Of the 75 probes, only four show strongly correlated and unexpected correlations: on 30 January at 05:25 UTC, less than 1 hour after the SEA-ME-WE4 cut, the test-boxes hosted by AMS-IX all switched to using the local node hosted by EMIX in Dubai, UAE. The situation lasted for more than seven days, until 20:00 (UTC), 5 February. These observations provide

positive confirmation that EMIX was reachable from Amsterdam during the entire cable outage period. However, the highly increased response times for the DNS queries do indicate congestion on the backup links. Also, the unfortunate leaking of the EMIX local node route announcements caused deteriorated service for those who received and preferred that route over a global node's announcement.

6.5. Analysis

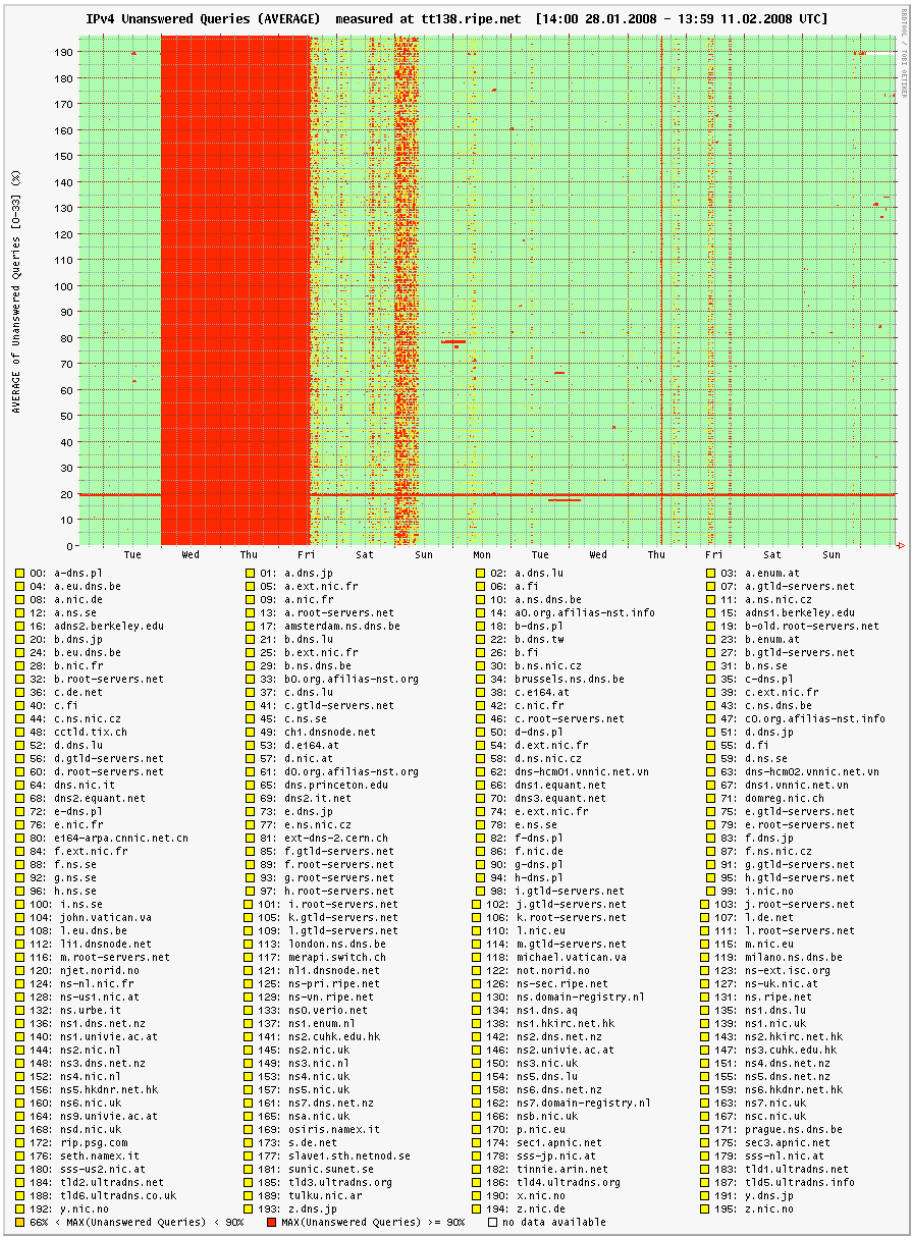


Figure 6.16: DNSMON graph for TT138. Each horizontal line corresponds to one monitored name server. Red dots mark unanswered queries.

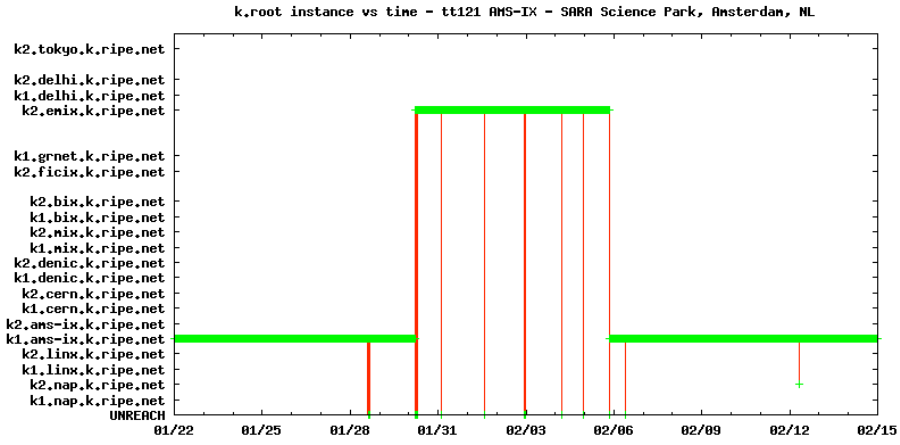


Figure 6.17: Results from DNSMON probe tt121: k.root instance used in a two-week time interval.

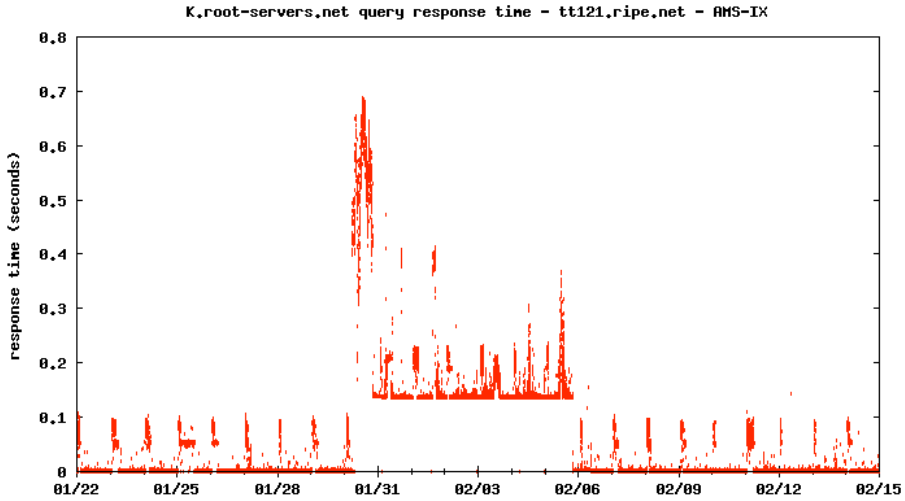


Figure 6.18: Results from DNSMON probe tt121: k.root query response time for TT121 in the same period.

6.6 Conclusions

Combining data from different measurement/monitoring systems, our analysis provides insight into how the cable outages affected Internet connectivity:

- Immediately following each cable cut, networks became unreachable, either because routes were withdrawn in BGP or because back haul links went down.
- Sites that had arranged for multiple transit providers observed massive rerouting in BGP, such as moving to satellite providers. Other sites were rerouted on the sub-IP level, moving to circuits set up over other, lower bandwidth or longer distance cable systems. Both types of back-ups experienced increased latencies and congestion, significantly impacting end users and likely causing instability in BGP.

The Mediterranean cable crisis demonstrates the importance of adequately dimensioned redundant connectivity, ideally following different geographical paths.

Chapter 7

YouTube Prefix Hijacking

On Sunday, 24 February 2008, Pakistan Telecom (AS17557) started an unauthorized announcement of the prex 208.65.153.0/24. One of Pakistan Telecoms upstream providers, PCCW Global (AS3491) forwarded this announcement to the rest of the Internet, which resulted in the hijacking of YouTube traffic on a global scale.

In this report we show how this event was observed by analyzing the data collected by the *Routing Information Service* (*RIS*) [roub] of *RIPE NCC* [ripa] and using publicly available tools developed by the *Compunet Research Group* of *Rome Tre University* [com].

The main results presented in this chapter are also described in [AKK+08] and it has also been published as a RIPE NCC's document at [ripf].

7.1 Event Time-Line

Before, during and after Sunday, 24 February 2008 AS36561 (YouTube) announces 208.65.152.0/22. Note that AS36561 also announces other prefixes, but they are not involved in the event.

Sunday, 24 February 2008, 18:47 (UTC) AS17557 (Pakistan Telecom) starts announcing 208.65.153.0/24. AS3491 (PCCW Global) propagates the announcement. Routers around the world receive the announcement, and YouTube traffic is redirected to Pakistan.

Sunday, 24 February 2008, 20:07 (UTC) AS36561 (YouTube) starts announcing 208.65.153.0/24. With two identical prefixes in the routing system, BGP policy rules, such as preferring the shortest AS path, determine which route is chosen. This means that AS17557 (Pakistan Telecom) continues to attract some of YouTubes traffic.

Sunday, 24 February 2008, 20:18 (UTC) AS36561 (YouTube) starts announcing 208.65.153.128/25 and 208.65.153.0/25. Because of the longest prex match rule, every router that receives these announcements will send the traffic to YouTube.

Sunday, 24 February 2008, 20:51 (UTC) All prex announcements, including the hijacked /24 which was originated by AS17557 (Pakistan Telecom) via AS3491 (PCCW Global), are seen prepended by another 17557. The longer AS path means that more routers prefer the announcement originated by YouTube.

Sunday, 24 February 2008, 21:01 (UTC) AS3491 (PCCW Global) withdraws all prefixes originated by AS17557 (Pakistan Telecom), thus stopping the hijack of 208.65.153.0/24. Note that AS17557 was not completely disconnected by AS3491. Prexes originated by other Pakistani ASes were still announced by AS17557 through AS3491.

7.2 Event Analysis

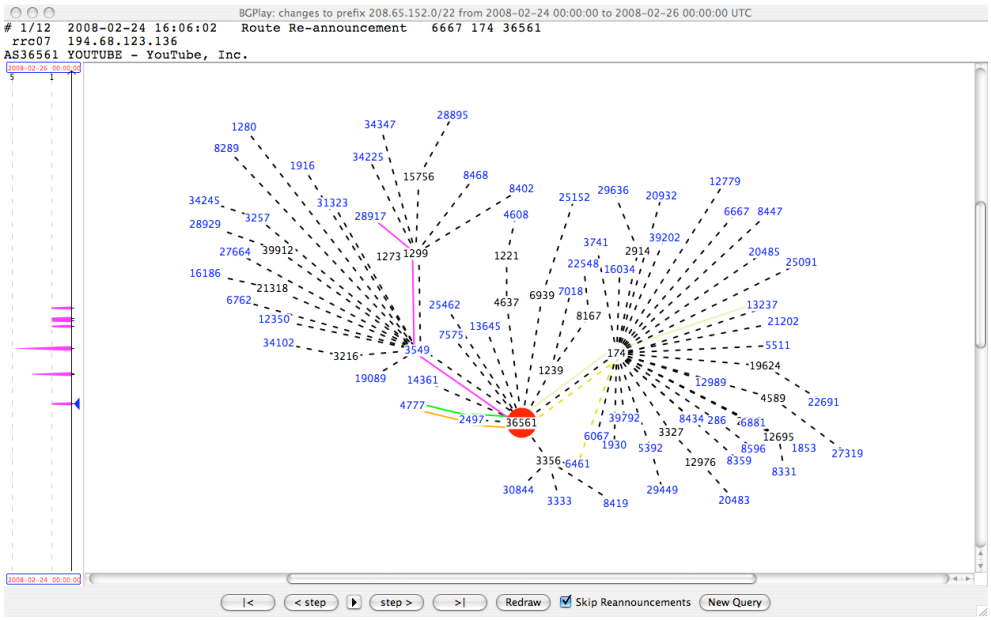
We now show how we analyzed the data collected by RISs collector peers, to understand the hijacking event, using some public available tools developed by the RIS and by the Compunet Research Group of Rome Tre University.

Pakistan aimed to block the YouTubes web site (youtube.com). youtube.com appears in the DNS with three distinct IP addresses: 208.65.153.238, 208.65.153.251

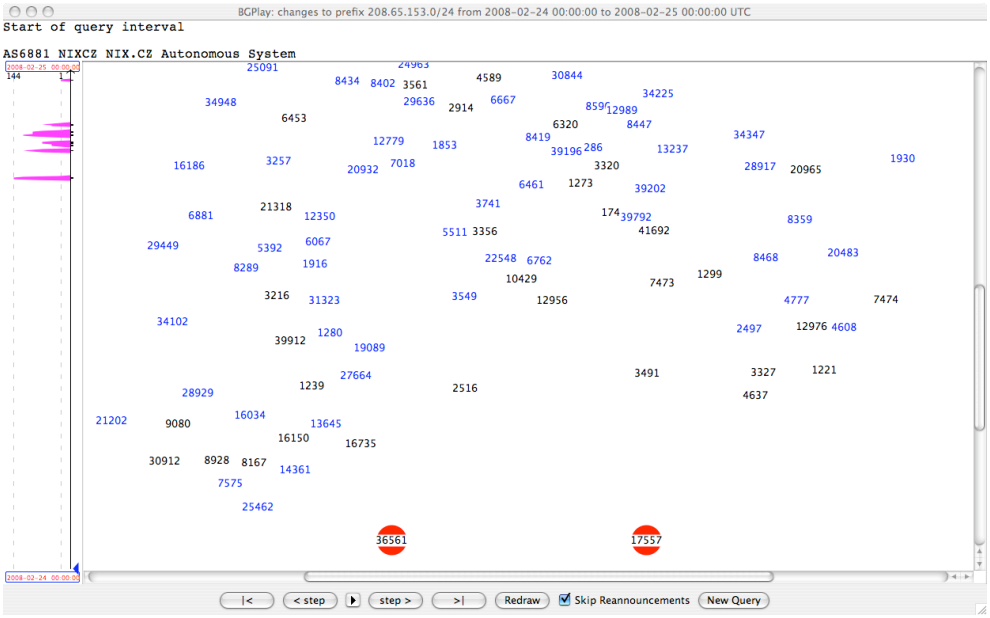
and 208.65.153.253. First, to find the prefixes originated by YouTube, we searched the routing table dumps of the various RIS's collector peers, by querying the RIS whois. This tool (accessible via whois protocol on [risb] or through the web interface at [risc]) provides a quick look at the most recent set of collector peers' routing tables. Once the hijacking was al- most over, RISwhois showed YouTube originating 208.65.152.0/22, 208.65.153.0/24 and 208.65.153.128/25. The /22 was the most widely seen prefixes (by 112 RISs collector peers). The /24 was observed by 105 peers. The /25 announcement, on the other hand, only reached 21 peers. Then, to have a more detailed view of the event, we looked at the BGP messages propagated through the Internet when the event occurred, using the RIS search tool [risa]. Searching for the period Sunday, 24 February 2008, 18:00 (UTC) to Monday, 25 February 2008, 01:00 (UTC), both AS17557 (Pakistan Telecom) and AS36561 (YouTube) resulted as origin of the /24 prex. Finally, to understand the dynamics of the route announcements, withdrawals and the competition in BGP between the Pakistani /24 and YouTube announcement, we used the visualization tools BGPlay [bgpb] and BGPath [bgpa]. These tools were designed and deployed by the Computer Networks Research Group and the former has been integrated into the RIS service portfolio. Next sections show BGPlay and BGPath snapshots illustrating the state of the network at some key points in time. It is important to note that all these tools can only show the BGP data collected by RISs collector peers and not routing, as such, for the whole Internet. Based on this information, it is not possible to make statements about how many sites had their traffic to YouTube hijacked.

7.2.1 Routing States - BGPlay Snapshots

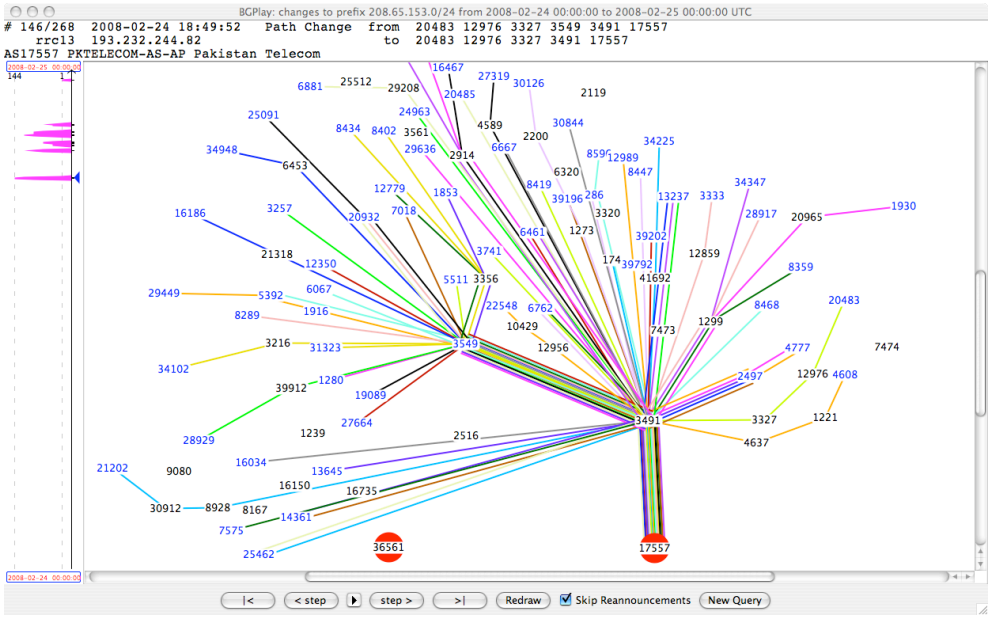
Before, during and after Sunday, 24 February 2008 AS36561 (YouTube) announces 208.65.152.0/22. Note that its connectivity almost doesnt change during the period of the hijacking. The prex 208.65.153.0/24 is not announced on the Internet before the event.



7.2. Event Analysis

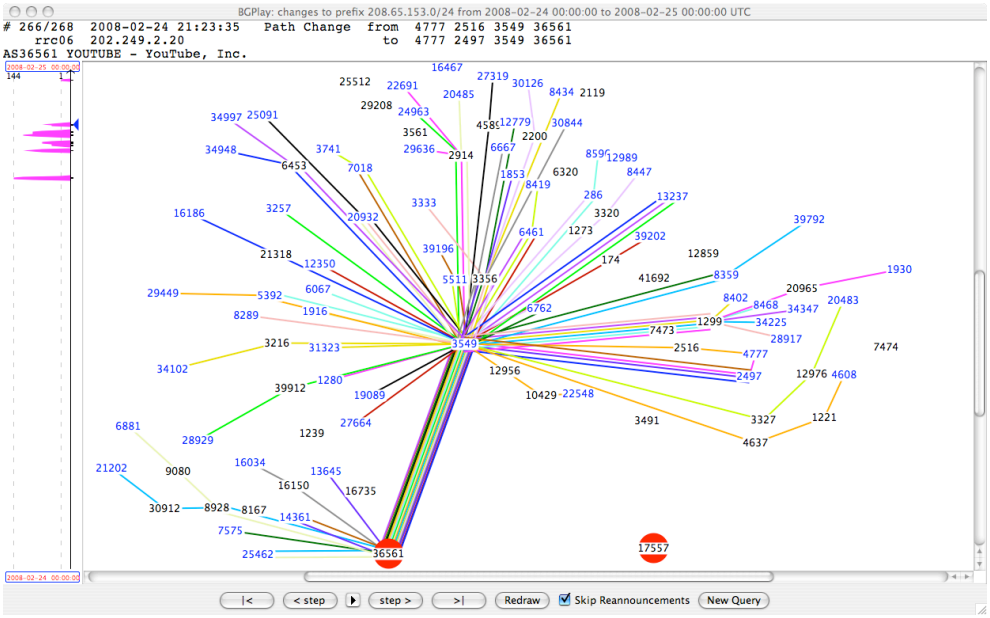


Sunday, 24 February 2008, 18:49 (UTC) AS17557 (Pakistan Telecom) has been announcing 208.65.153.0/24 for the past 2 minutes. The RISs collector peers around the world have received the route update. The YouTube traffic is being redirected to Pakistan.

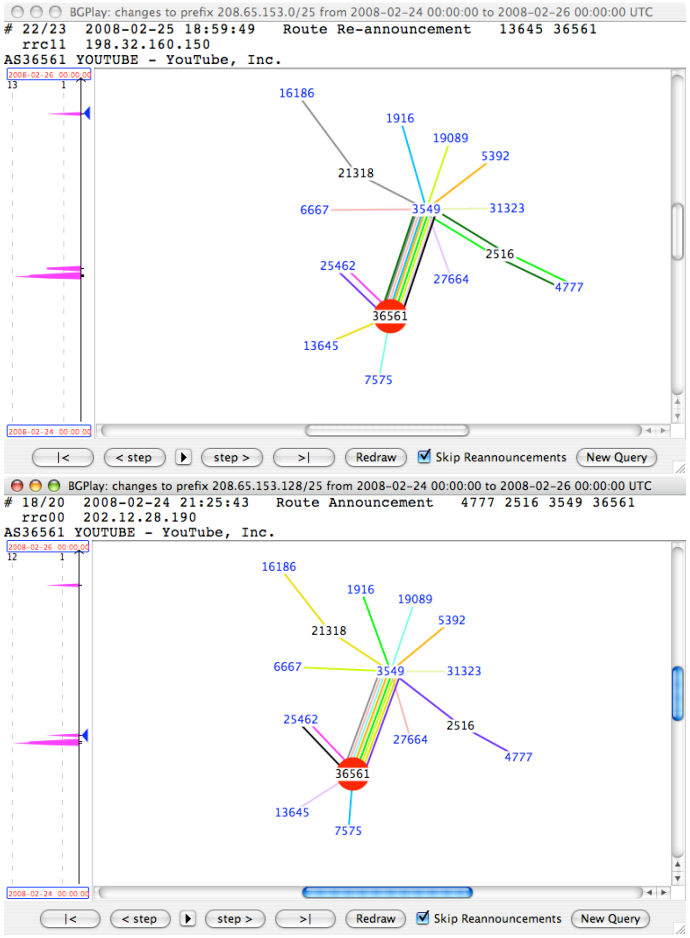


7.2. Event Analysis

Sunday, 24 February 2008, 21:23 (UTC) AS36561 (YouTube) has been announcing 208.65.153.0/24 since 20:07 (UTC). The bogus announcement from AS17557 (Pakistan Telecom) has been withdrawn, and RIS peers now only have routes to YouTubes AS36561.



Since Sunday, 24 February 2008, 20:18 (UTC) AS36561 (YouTube) are announcing 208.65.153.0/25 and 208.65.153.128/25. Note that both of these prefixes are much less visible on the Internet than the /24 prex.



7.2.2 Path Evolution of the Hijacked Prefix as Observed by a RIS Peer - BGPath Snapshots

In order to have a complete view of the routing changes that the hijacked prex (208.65.153.0/24) underwent over the course of the hijacking, we looked at the path evolution over time of the hijacked prex. Figure 7.1 shows the evolution of the path chosen by a specific peer (in this case AS3333, RIPE NCC) to reach the hijacked prex. Namely, Figure 7.1 shows that:

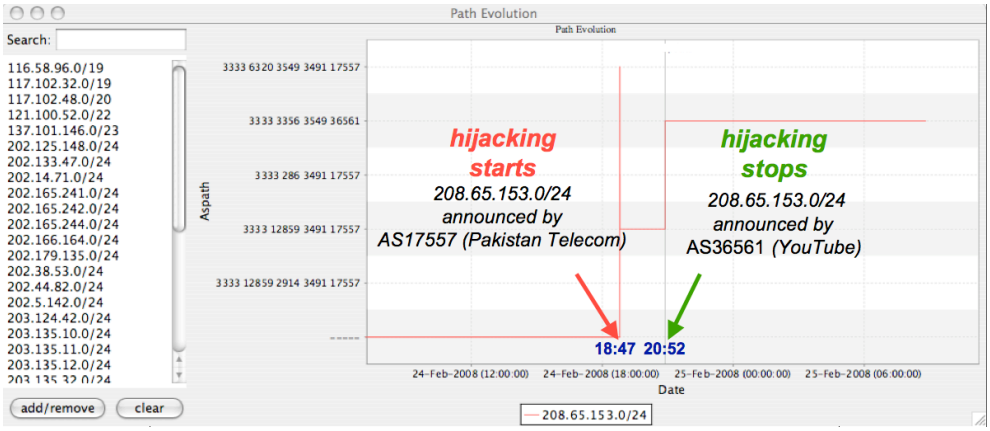


Figure 7.1: Path evolution of the hijacked prex.

Until Sunday, 24 February 2008, 18:47 (UTC) AS3333 (RIPE NCC) had no path toward 208.65.153.0/24.

On Sunday, 24 February 2008, from 18:47 to 20:52 (UTC) AS3333 (RIPE NCC) observed 208.65.153.0/24 being announced by AS17557 (Pakistan Telecom) through two distinct paths (3333 6320 3549 3491 17557 and 3333 12859 3491 17557).

Since Sunday, 24 February 2008, 20:52 (UTC) AS3333 (RIPE NCC) has observed 208.65.153.0/24 being announced by AS36561 (YouTube) through the path 3333 3356 3549 36561.

7.3 Conclusions

As the above timeline shows, this event happened in a relatively short time interval: YouTube reacted about 80 minutes after the Pakistan Telecom announcements, and all the major events finished after about two hours. While this report showed that the tools provided by RIPE NCC (such as RISwhois [risb] and BGPlay [bgpb]) can help in following and analyzing events even on such a short timeline, we also note that unauthorized announcements like this can be prevented from spreading throughout the Internet by appropriate routing configuration by operators of Autonomous Systems. The RIPE NCC provides the RIPE Routing Registry [regc] in order to facilitate such configuration. Currently the RIPE community is discussing the introduction of digital certificates for Internet number resources. These certificates are intended to provide a tool to further enhance routing configuration throughout the Internet.

Conclusions

As the Internet grows far beyond the initial expectations, interdomain routing dynamics become more and more complex and distributed monitors collect larger and larger amounts of BGP data, it becomes harder and harder to understand the Internet dynamics. Thus, both researchers and network operators need methodologies to effectively detect and analyze network events and tools to efficiently process BGP data.

This thesis tackles these problems exploring various directions.

First, we defined a flow-based system to model the BGP dynamics and, based on this model, a methodology to identify root causes of interdomain routing events. Our approach adopts the point of view of an ISP interested in monitoring and debugging the reachability of its own prefixes and correlates it with the status of the global routing. Namely, we start the analysis from a specific BGP change and we try to track back its cause using both single and cross-monitor perspectives. We evaluated our methodology through Internet scale simulations and analysis of real world data. Experimental results show that our approach usually fairly accurately locates the portion of Internet responsible for the given routing change. In order to support the methodology, we developed BGPath, a publicly available tool that collects BGP data from about 800 monitors spread over the Internet by the RIS and RV projects, efficiently estimates the usage of inter-AS peerings and effectively visualizes detailed and aggregated routing information related to user-specified prefixes. Exploiting stream-like algorithms, BGPath provides the user with nearly-real time information.

On the other hand, we also embraced the "global" perspective and we proposed a new technique to identify major contributors to the interdomain routing dynamics, by using the Principal Component Analysis over BGP data collected by distributed vantage points. Through simulations and real case studies we showed that this new methodology can pin down routing changes to spe-

cific ASes or links. Also, since different vantage points have different views of the Network, the application of PCA enables us to separate unrelated routing events.

Since RIS' and RV's BGP data only show active routing configurations, we also studied the interdomain static network configurations contained in the Internet Routing Registry to get a more complete view of the Internet topology. Namely, we defined a methodology to extract peering relationships from the IRR and we developed an on-line service which implements our methodology. Even if the routing registries contain out-of-date data, we found about 10 times more inter-AS peerings than what we can observe from RIS' and RV's monitors. Overall, comparing static BGP configurations against actual routing data significantly helps understand the Internet dynamics and predict their evolution in response to network events.

In order to correctly interpret interdomain routing data, we also studied the route diversity that can be observed from remote vantage points and we defined a methodology that extracts diversity relationships from a continuous stream of BGP updates. We thus characterized the route diversity against the Internet customer-provider hierarchy and we studied its sensibility to the choice of vantage points.

We finally applied the approaches previously described to analyze real network events, in order to show their effectiveness.

Appendices

Mediterranean Fiber Cut: Case Studies

Case Study 1 - Unreachable Prefixes From BGP Point of View (Egyptian Prefix)

Introduction

Due to the fibre cuts, some prefixes were unreachable for a significant time period. We analyzed the effects of the FEA fibre cut on the prefixes originated by AS20484 (Yalla Online, Egypt) (Yalla Online, Egypt).

Routing States of a Prefix Originated by AS20484 - BGPlay screenshots

We looked at the routing dynamics of the prefix 196.20.62.0/24 (originated by AS20484), using BGPlay. The following figures show some of the key routing changes the prefix underwent. Overall, the prefix was completely disconnected from the Internet for about 11 days.

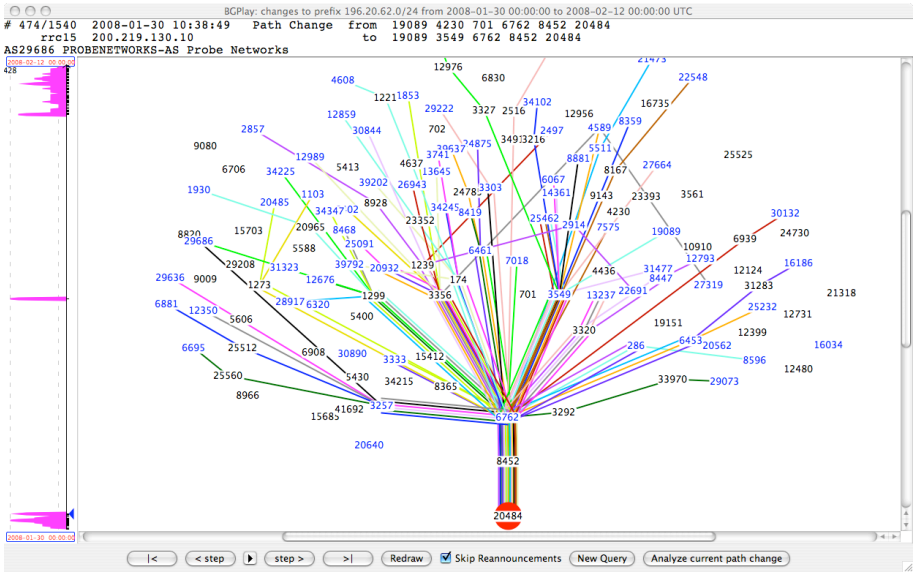


Figure 1: **10:38 (UTC), 30 January 2008** Some hours after the fibre cut, the prefix is still reachable by most of the RIS peers.

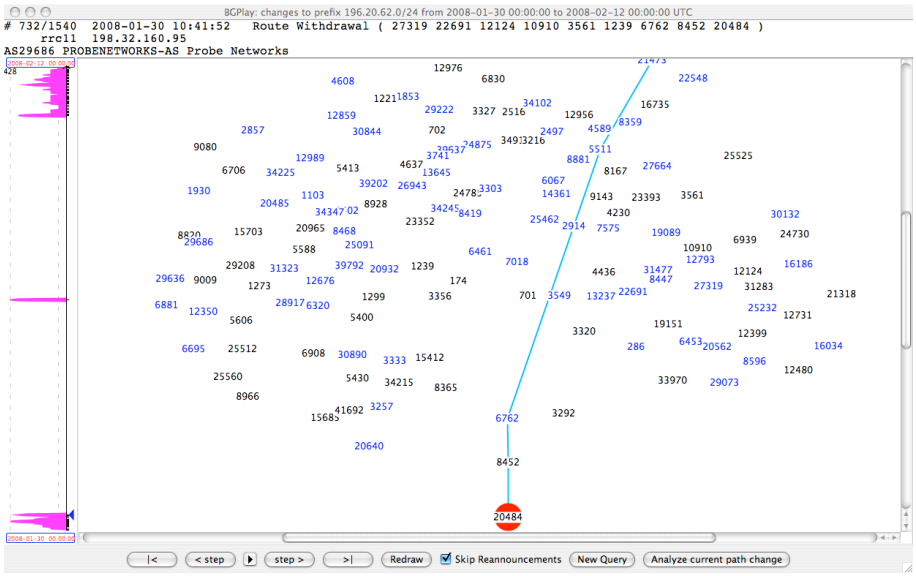


Figure 2: **10:41 (UTC), 30 January 2008** In a few minutes the prefix has lost all its paths towards RIS peers. The path still shown connecting AS20848 and AS21475 (ISP Global Ukraine LAN Lviv, Ukraine) is the last to go. It was withdrawn some seconds after the time of this snapshot.

Case Study 1 - Unreachable Prefixes From BGP Point of View (Egyptian Prefix)

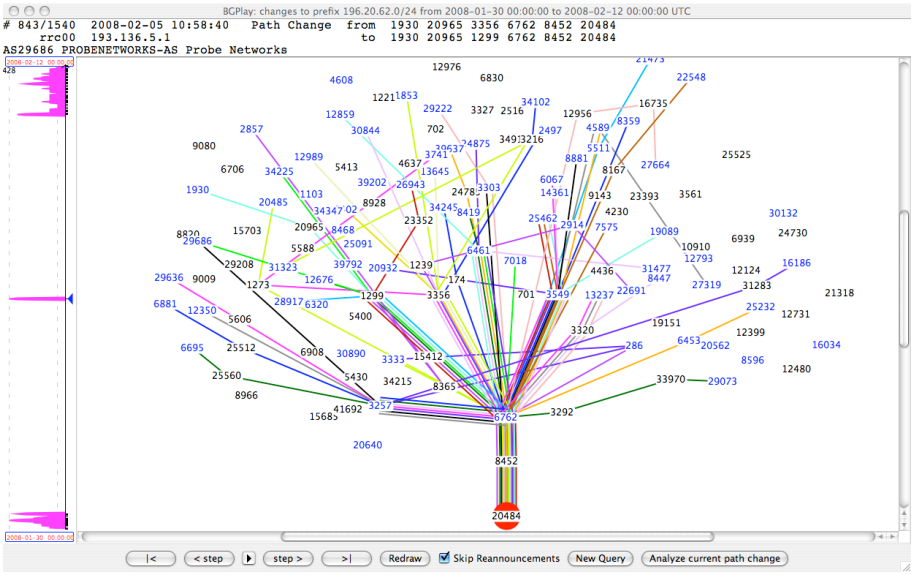


Figure 3: **10:58 (UTC), 5 February 2008** For a very short time period, the prefix temporarily regained its connectivity to the Internet.

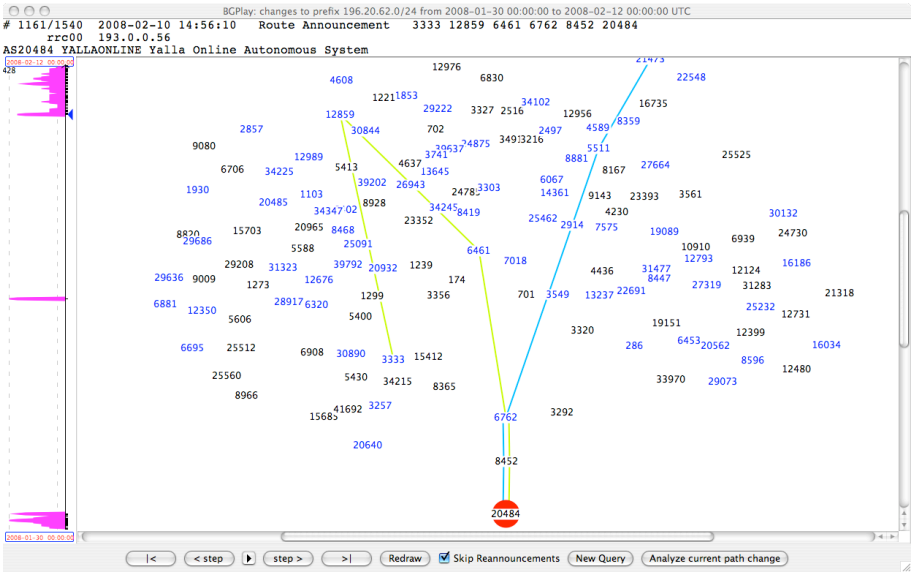


Figure 4: **14:56 (UTC), 10 February 2008** First signs of recovery, route announcements start to arrive at the RIS peers.

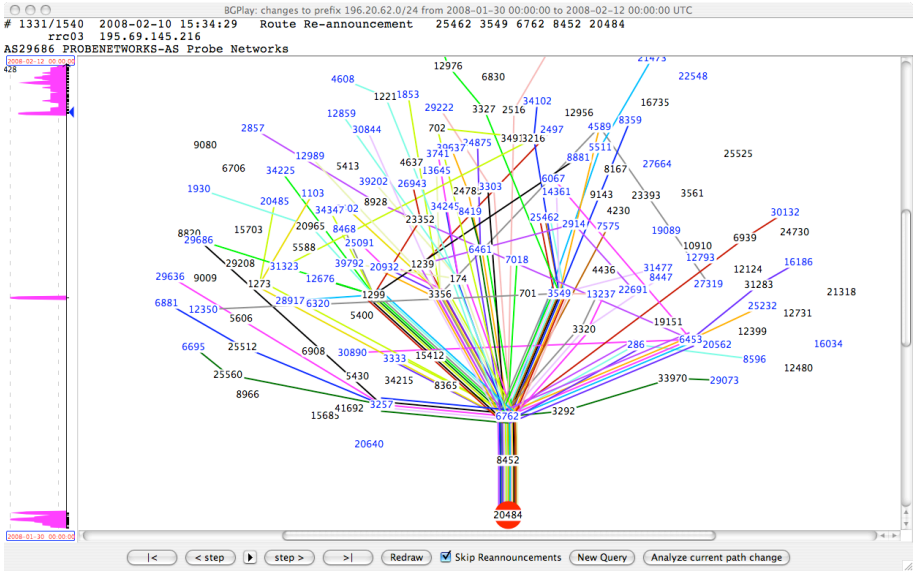


Figure 5: **15:34 (UTC), 10 February 2008** All the RIS peers are observing the prefix again.

Case Study 1 - Unreachable Prefixes From BGP Point of View (Egyptian Prefix)

Cross Prefix Analysis - BGPath screenshots

We also looked at the routing dynamics of all the prefixes originated by AS20484, and other prefixes' usage of the AS20484-AS8452 link, using the BGPath tool.

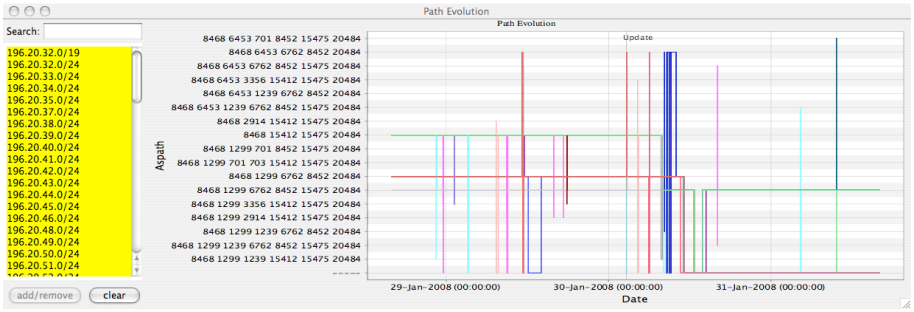


Figure 6: **Path Evolution of All the Prefixes Originated by AS20484** All the prefixes originated by AS20484 and using AS8452 (TEDATA) as upstream provider either changed their paths or completely lost their connectivity to the Internet for the whole period.

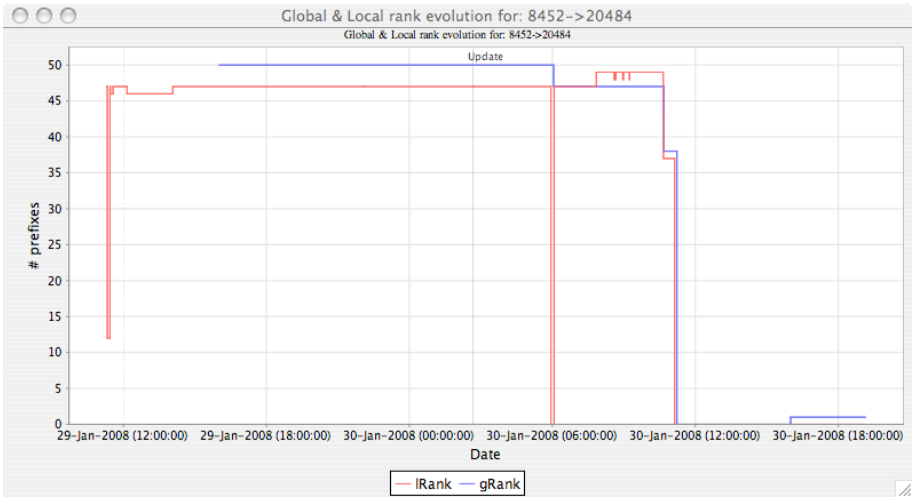


Figure 7: **Evolution of the Number of Prefixes on Specific AS-links** The AS-link 20484-8452 usually carries about 50 prefixes. The figure below shows that all these prefixes stopped using the link during the morning of 30 January 2008. Because the timing of that event does not align with the known start times of the cable outages, the link may have been shutdown by human operator intervention.

Conclusions

This analysis shows that from 30 January 2008 to 10 February 2008, the AS-link 20484-8452 experienced some major event. All the prefixes usually passing through this link (originated by AS20484 and other ASes) either were unreachable or changed their routes for most of the period.

Case Study 2 - BGP Still Carries Routes While Traffic is Black Holed (Bahrain)

Introduction

As shown in Section 6.5.2.1, the TTM box in Manama, Bahrain was not reachable by any other TTM box between 30 January and 2 February because of the outage on the SEA-ME-WE-4 cable. If we look into RIS routing tables there is no indication of an outage, instead route announcements were rather stable for the prefixes originated by AS35313 (2Connect), which hosts the test-box.

Routing States of a Prefix Originated by AS35313 - BGPlay Screenshots

AS35313 announces two prefixes (80.88.240.0/20, 80.88.244.0/22), and both of them underwent almost the same routing changes during the fibre outage time period. Therefore, we only show some key routing changes of one prefix (80.88.240.0/20) with the following BGPlay screenshots.

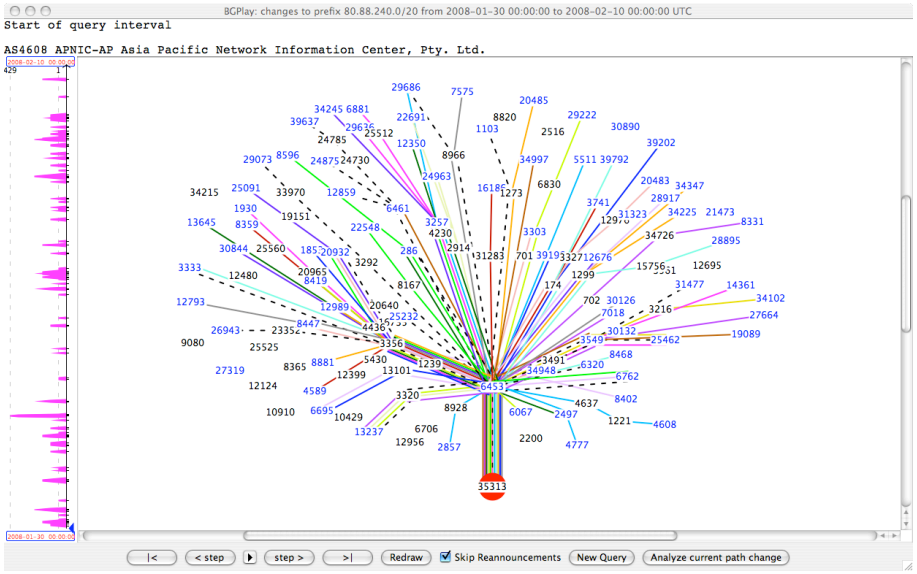


Figure 8: **00:00 (UTC), 30 January 2008** Before all the fibre outages.

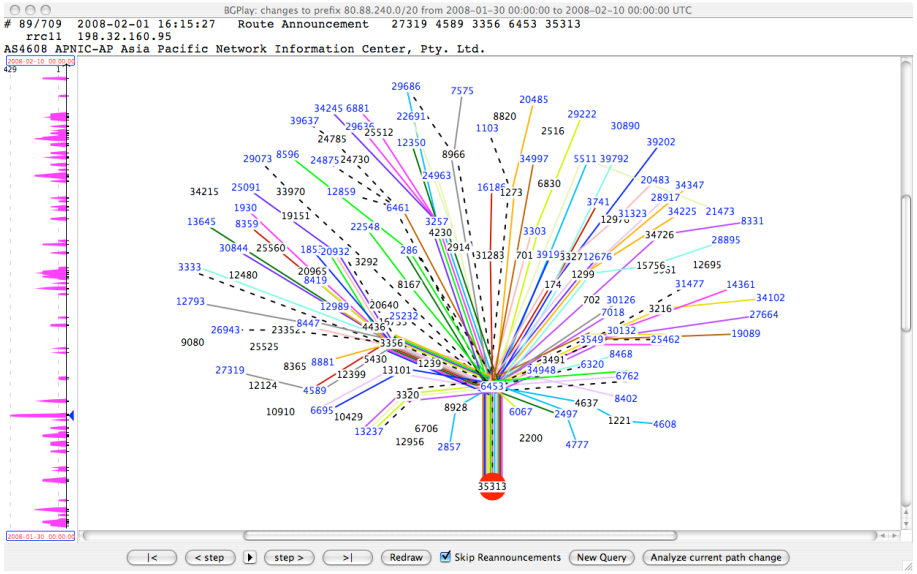


Figure 9: **16:15 (UTC), 1 February 2008** More than a day after the last the fibre outage, the paths are almost the same. AS35313 starts losing some paths.

Case Study 2 - BGP Still Carries Routes While Traffic is Black Holed (Bahrain)

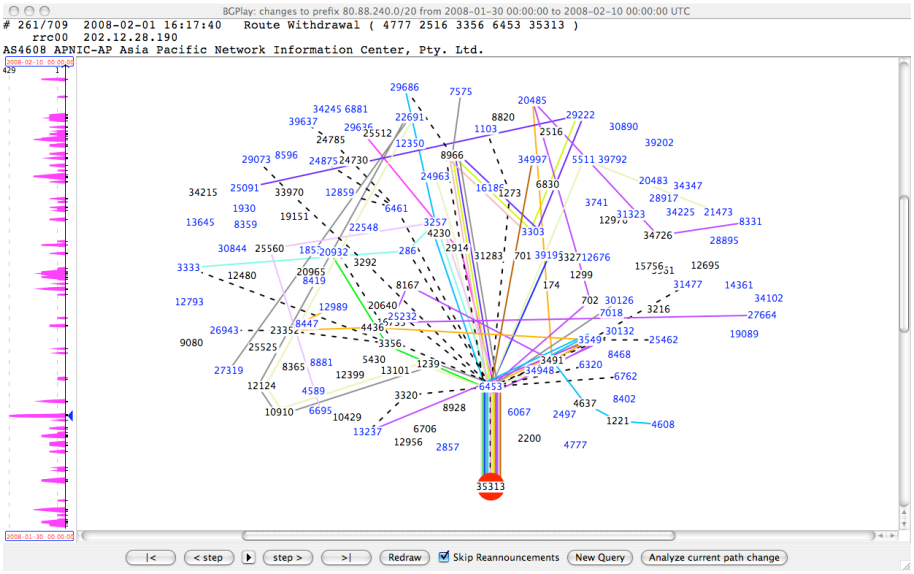


Figure 10: **16:17 (UTC), 1 February 2008** In two minutes, AS35313 reaches the most disconnected status during the fibre cut period.

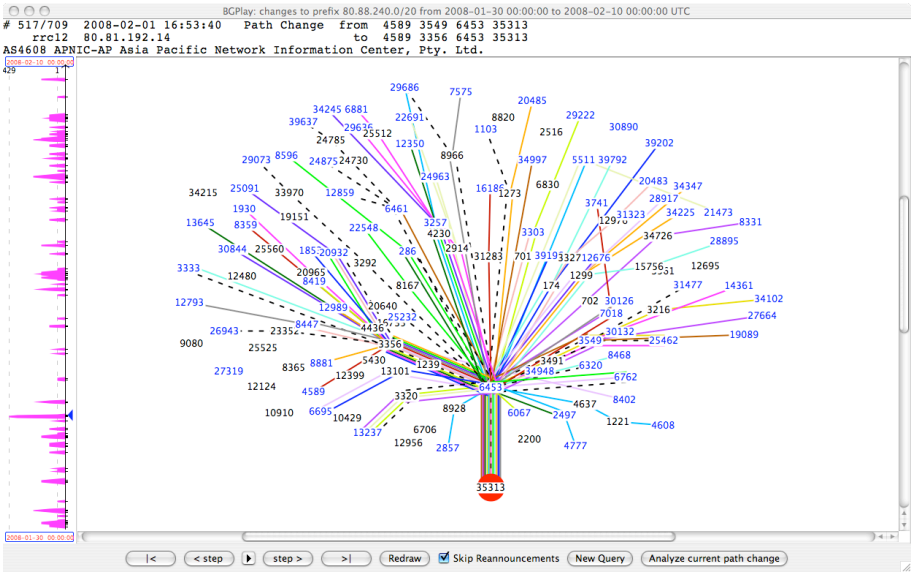
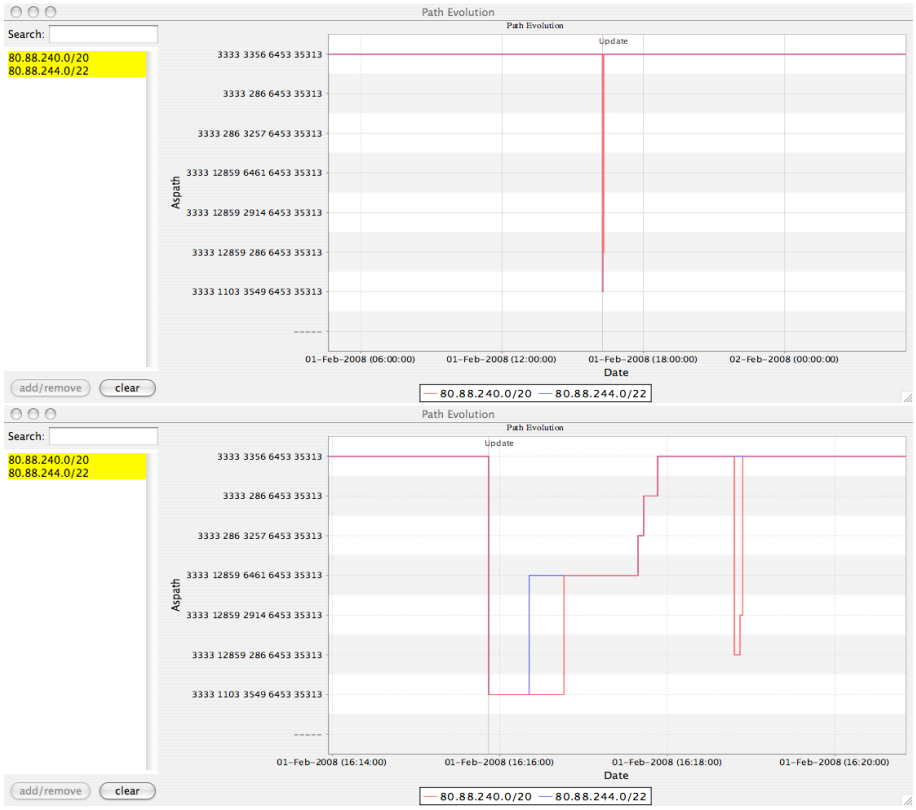


Figure 11: **16:53 (UTC), 1 February 2008** About 30 minutes later, practically all of the paths have been recovered. Most likely this short-lived event was caused by operator intervention (a manual router reset or similar event). The results from TTM traceroutes indicate that end-to-end connectivity was restored another 35 minutes later, around 17:29 (UTC).

Path Evolution of All the Prefixes Originated by AS35313 - BGPath Screenshots

From the point of view of peers at RRC00 (RIPE NCC) and RRC03 (AMS-IX), AS35313's prefixes were reachable at all times during the cable outage period. The routes were also stable almost all time, except for about five minutes when both prefixes experienced some path changes. The following BGPath screenshots illustrate this.



Conclusions

The TTM data show that test-box 138 in Bahrain had no connectivity to any other TTM test-box for 2.5 days, starting 04:30 (UTC), 30 January. However during this period, hardly any changes were seen in BGP for the prefixes originated by AS35313 (the site which hosts the box). This shows that the presence of a route in BGP is no guarantee of a working Internet connection.

As explained in Section 6.5.2.1, we suspect the prefix was (statically) originated by a router located in London, hosted or owned by Teleglobe. Thus the

failure of the submarine cable did not trigger a withdrawal of the prefix from BGP routing tables. Later, the Teleglobe-Bahrain traffic usually carried by the SEA-ME-WE4 cable was rerouted via a different submarine cable. Because this had no effect on how the prefix was announced in BGP, the RIS collectors see no changes in ASpath.

Case Study 3 - BGP Rerouting of Prefixes

Introduction

Due to the cable outages, some prefixes changed their routes for a significant time period. We analyzed the effects of the SEA-ME-WE4 outage on the prefixes originated by AS17641 (Infotech, Bangladesh).

Routing States of a Prefix Originated by AS17641 - BGPlay Screenshots

We looked at the routing dynamics of the prefix 202.65.10.0/23 (originated by AS17641), using BGPlay. The following figures show some of the key routing changes the prefix underwent.

Case Study 3 - BGP Rerouting of Prefixes

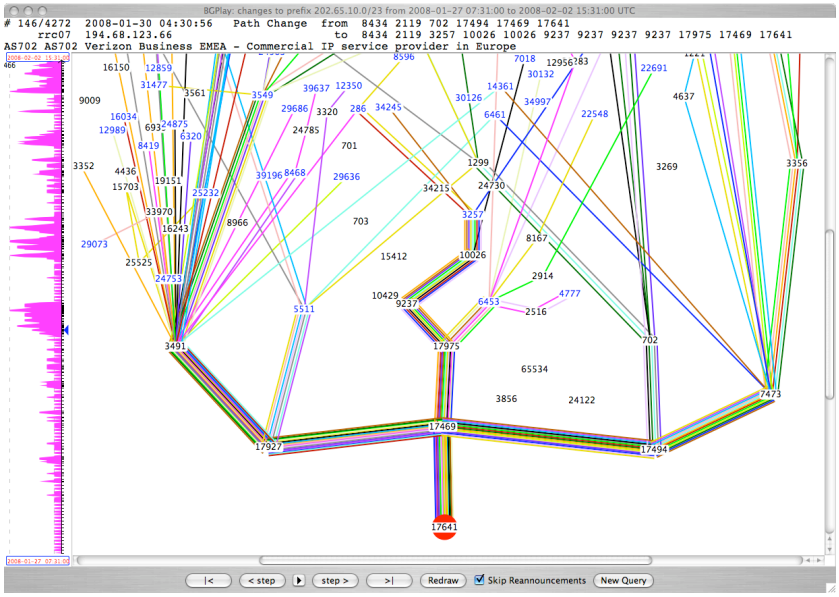


Figure 12: **4:30 (UTC), 30 January 2008** Until this moment, not much has happened in terms of BGP messages for this prefix. The purple histogram at the left of the graph shows we are at the start of a period with high activity, triggered by one of the cable faults.

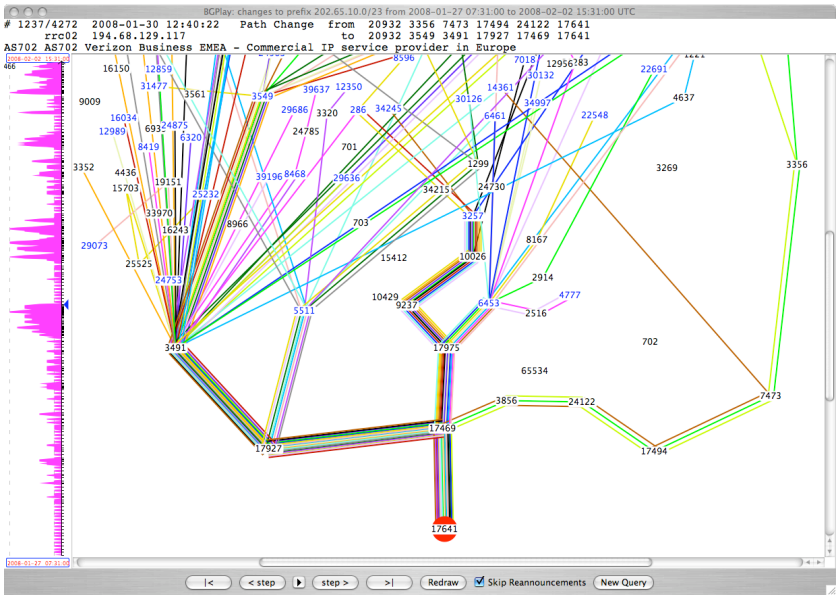


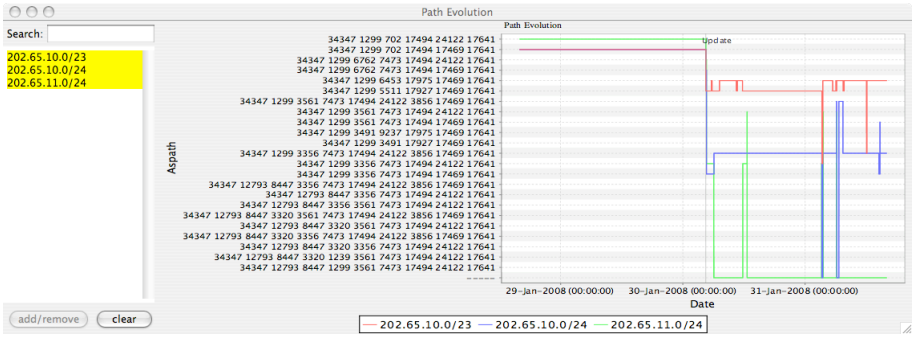
Figure 13: **12:40 (UTC), 30 January 2008** Five hours after the event, BGP temporarily stabilizes. Note that AS702 (UUNET Europe) is no longer seen in any route to AS17641. Also, only a few routes remain using the links through AS17494 (Bangladesh Telegraph and Telephone Board). All others switched to using AS17927 (WEBSATMEDIA PTE LTD, Satellite Over IP, Singapore) and AS17975 (APT Telecom Services Ltd., Hong Kong) for transit.

Cross Prefix Analysis - BGPath Screenshots

We also used the BGPath tool to look at the routing dynamics of all prefixes originated by AS17641 as well as the dynamics of some inter-AS links seen in the AS paths ending at AS17641.

Path Evolution of All the Prefixes Originated by AS17641 AS17641 announces three prefixes.

The prefixes (202.65.10.0/23 and 202.65.10.0/24), whose paths usually pass through AS17494, have been rerouted through AS17927 and AS17975. The prefix 202.65.11.0/24, which passes through AS24122 (BDCOM Online Limited, Bangladesh), completely lost its connectivity.

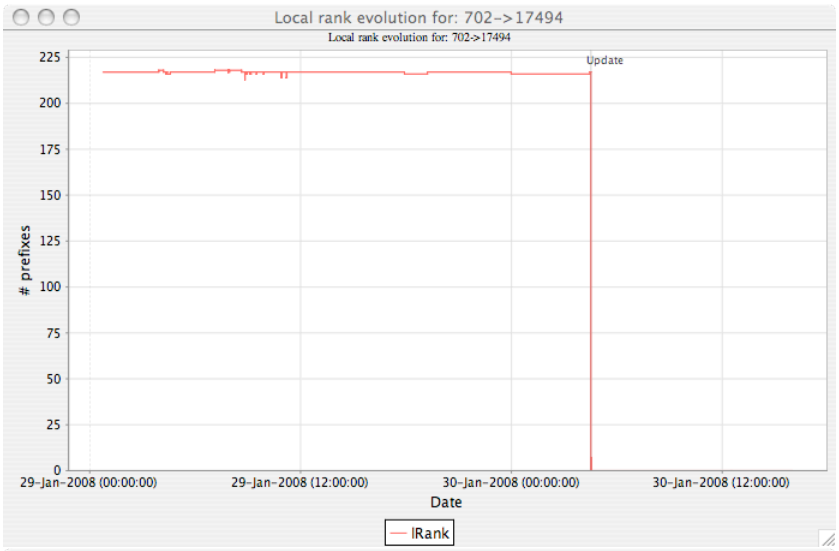


Evolution of the Number of Prefixes on Specific AS Links We analyzed how the number of prefixes through AS links "related" to AS17641 (that is, adjacent to AS17641's upstream providers and announcing AS 17641's prefixes) changed.

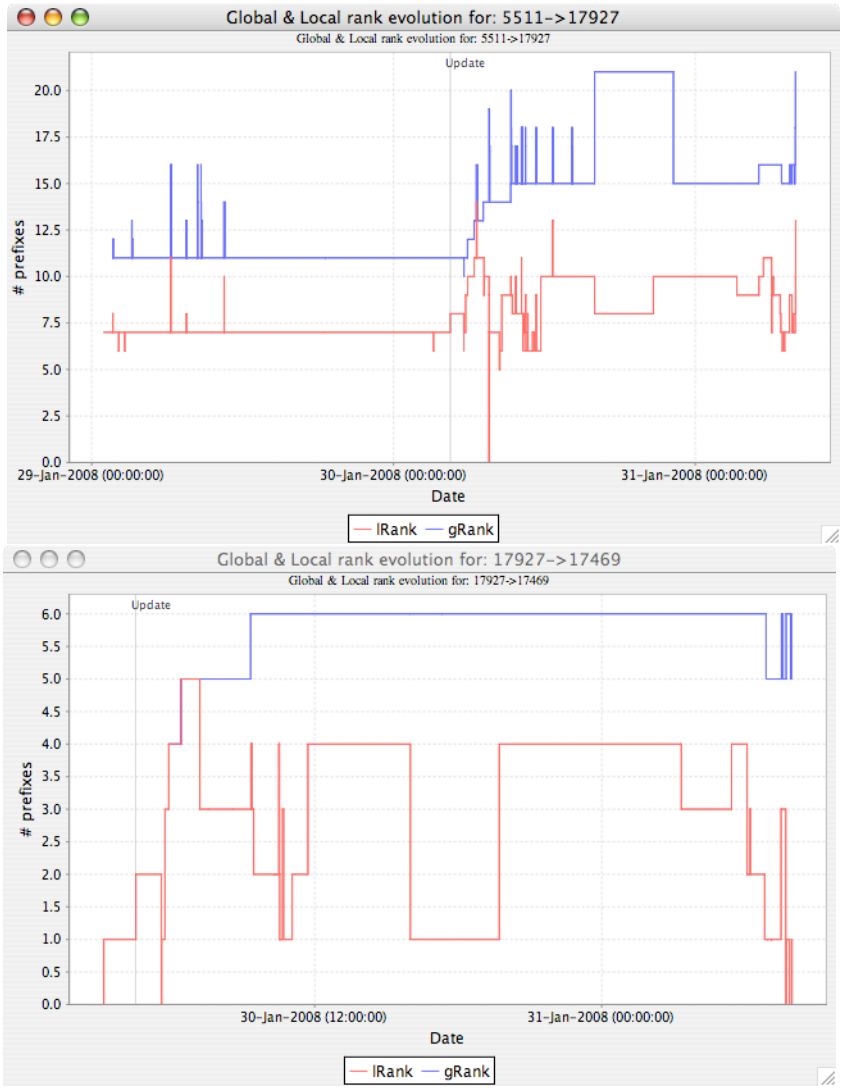
30 January 2008 Fault

The BGPlay plots showed how AS17641 stopped using paths through 702-17494-17469 after the cable cut. At the same time, many other prefixes stopped being routed over these paths too. The following figures show how the number of prefixes announced by AS17641 and others through the AS links 702-17494 and 17494-17469 suddenly dropped. In particular, the AS link 702-17494, usually used by more than 200 prefixes, is not used by any of them for a significant time period. This is evidence of a major network event, something that affects the connectivity between AS702 and AS17494.

Case Study 3 - BGP Rerouting of Prefixes

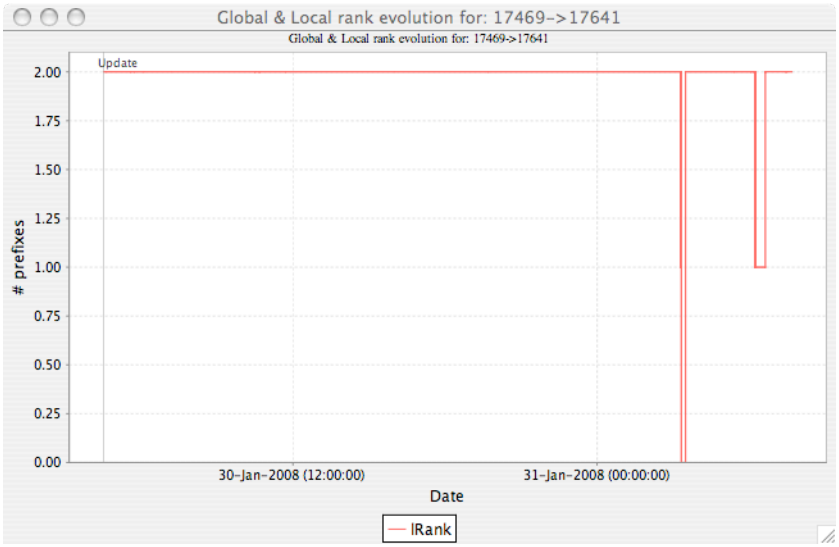


The BGPlay plots also showed that, as a consequence, AS17641's prefixes moved from 702-17494-17469 to 5511-17927-17469. Other prefixes also changed paths, preferring these two Autonomous Systems after the fault. As illustrated in the figures below, the number of prefixes announced by AS17641 and others through the AS links 5511-17927 and 17927-17469 increased.



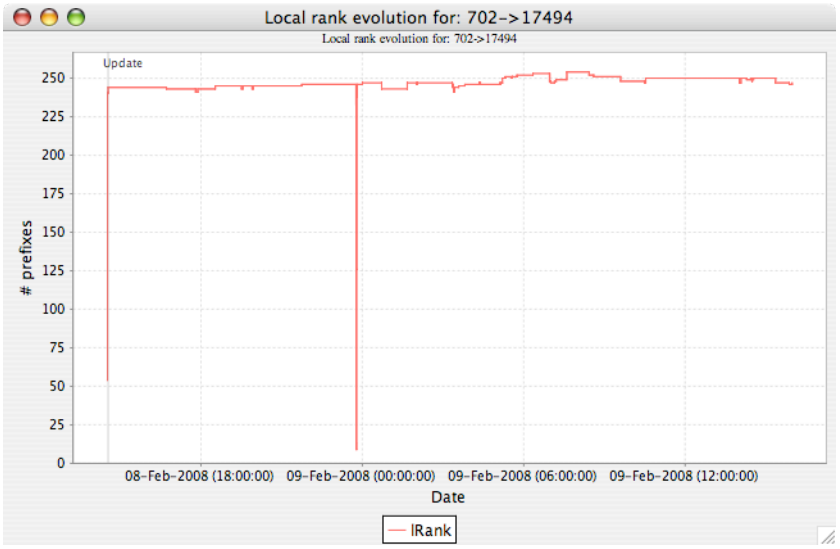
Case Study 3 - BGP Rerouting of Prefixes

Figure 14: Because almost all the prefixes affected by the event changed their paths without losing their connectivity, almost no change was observed on the AS link 17469-17641.



8 February 2008 Recovery

When the SEA-ME-WE4 cable was repaired and the original routing state recovered, we observed opposite changes in the number of prefixes on the AS links. As shown in the figure below, the link 702-17494 sees a sudden increase to it's old level of about 250 routed prefixes in the afternoon of 8 February 2008.



Conclusions

This analysis shows that from 30 January 2008 to 8 February 2008, the AS-link 702-17494 experienced some major event, and all the prefixes (whether originated by AS17494 or not) usually passing through this link were rerouted through different links. The timing of this event, 04:30 (UTC), aligns well with the recorded outages of the TTM box in Bahrain and other events in BGP. Because the FEA cable went down at 08:00, we conclude the AS link 702-17494 is set up over the SEA-ME-WE4 cable.

Case Study 4 - OmanTel: Explosion in AS Path Count, Hours of BGP Churn

The AS path changes graph for Oman caught our attention as being quite different from the general pattern: instead of going down, because of loss of connectivity, the number of distinct AS paths, as observed from RRC03 peers in the eight hourly RIB dumps, went up. The average AS path length also increased for the duration of the cable outages.

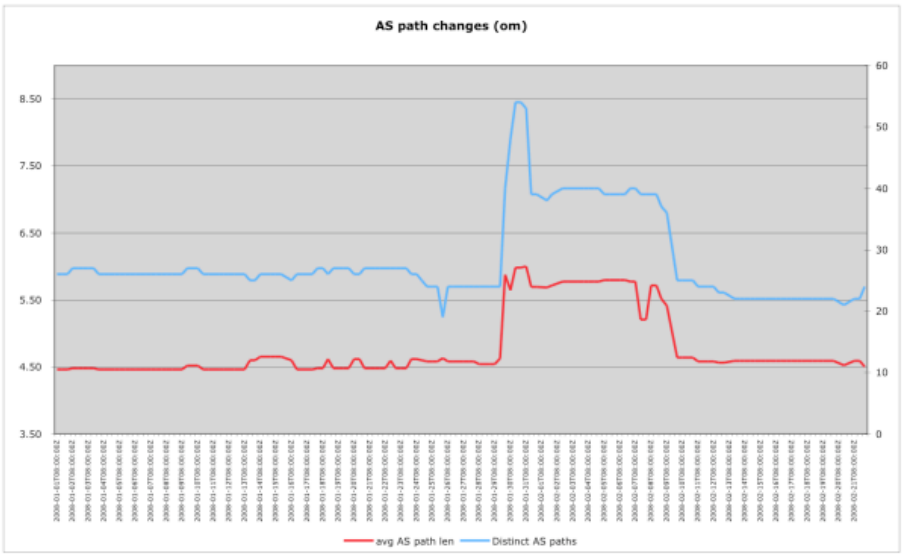


Figure 15: Number of AS Paths And Average Path Length For Oman.

The RIR stats files show only one AS assigned directly to a provider in Oman: AS28885 - OmanTel NAP. In January 2008, RIS saw 26 prefixes originated by this AS. Looking at the raw data, we noticed these prefixes are usually announced to the RIS peers in batches. One BGP update message carries the bulk of OmanTel prefixes, one or two other updates carry the rest. From RIS data alone is not possible to deduce with 100% certainty the reasons behind the observed behavior; however, we imagine specific routing policies related to the networks served by the prefixes could play a role.

In relation to the cable outages we see the following:

Before the cable fault Each collector peer received the same AS path in all update messages for AS28885. This indicates the same routing policy for all prefixes.

During the cable outage The origin AS apparently used a different policy for different sets of prefixes. So the collector peers receive different AS paths on the update messages for the full set of Oman prefixes.

Routing States of a Prefix Originated by AS28885 - BGPlay Screenshots

We looked at the routing dynamics of the prefix 82.178.64.0/18 (originated by AS28885), using BGPlay.

Case Study 4 - OmanTel: Explosion in AS Path Count, Hours of BGP Churn

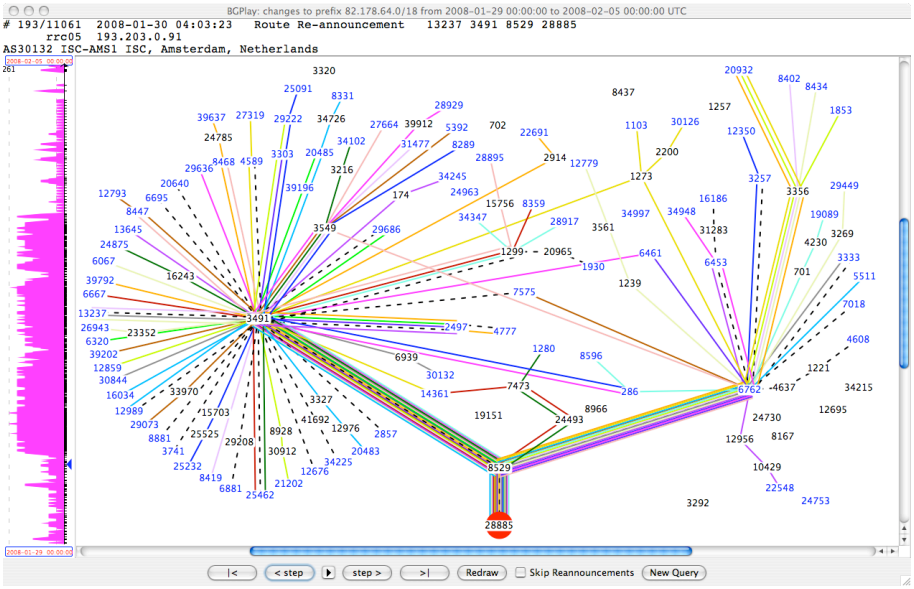


Figure 16: **04:03 (UTC), 30 January 2008** Before the cable outages; primary transits for Oman are AS3491 (PCCW) and AS6762 (Telecom Italia Sparkle). Note how the purple histogram on the left indicates a period of prolonged, continuous BGP updates is about to begin. Over 10,000 messages were recorded in 90 hours, which means that on average the collective of all RIS peers saw an announcement or withdrawal every 30 seconds.

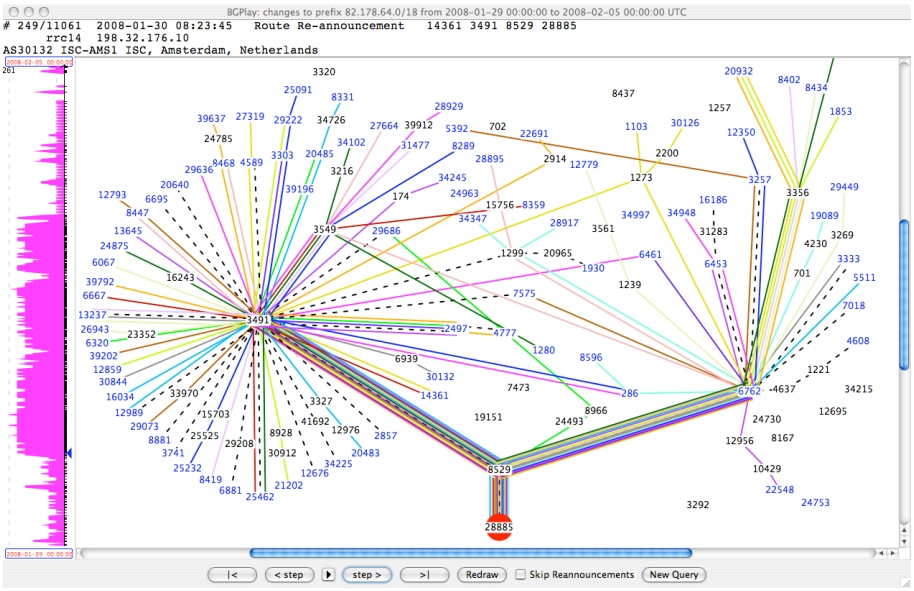


Figure 17: **08:23 (UTC), 30 January 2008** Shortly after the second cable went down; first signs of rerouting are visible.

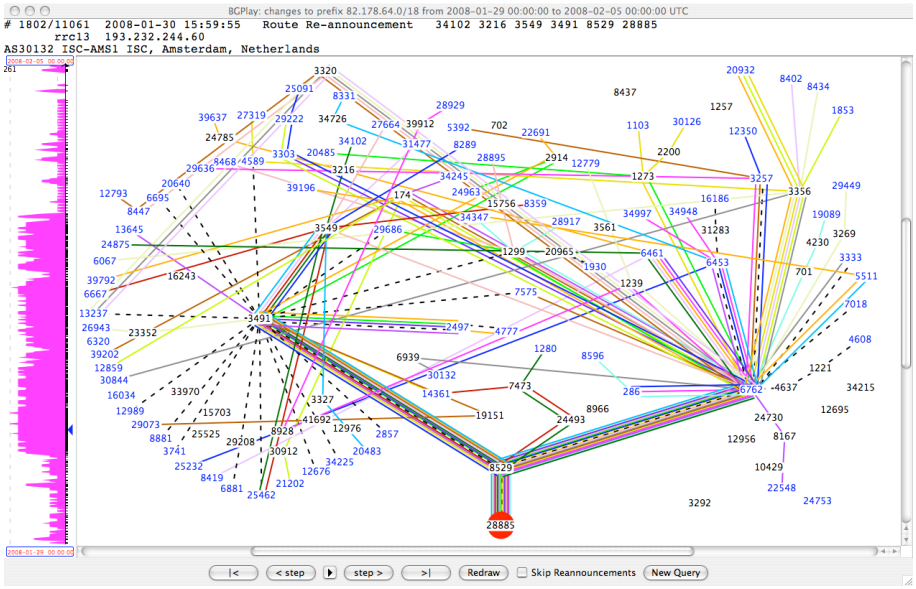


Figure 18: **15:59 (UTC), 30 January 2008** Immediately before the second RIB dump of the day. Many peers have switched from using AS3491 to AS6762 as transit to Oman. From the graph we can see most of these peers need more AS hops to reach AS28885, thus the average AS path length increases.

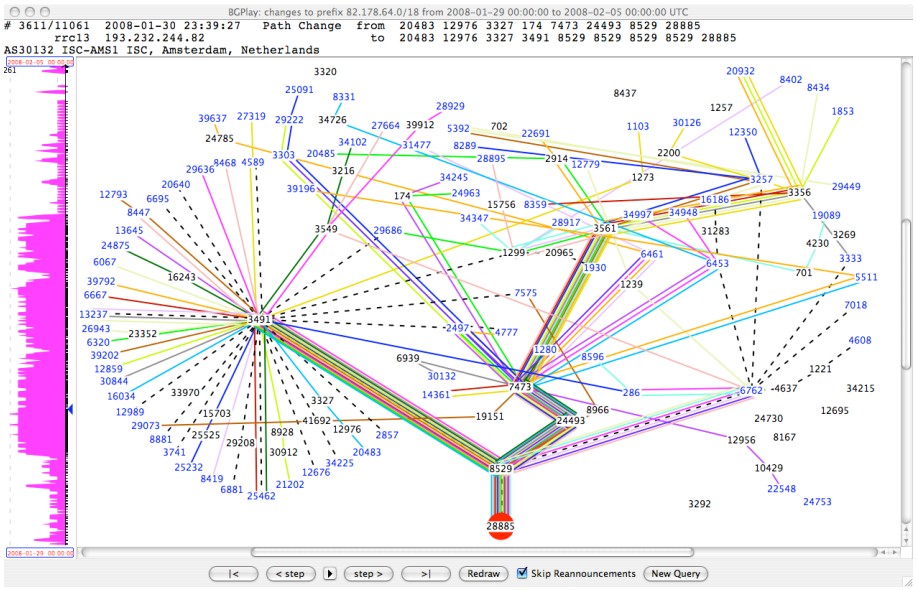


Figure 19: **23:39 (UTC), 30 January 2008** Before the last RIB dump of the day. AS24493 (STIXLITE Transit Service Provider Singapore) has taken over the transit for most peers who first used AS6762 (Telecom Italia Sparkle).

Case Study 4 - OmanTel: Explosion in AS Path Count, Hours of BGP Churn

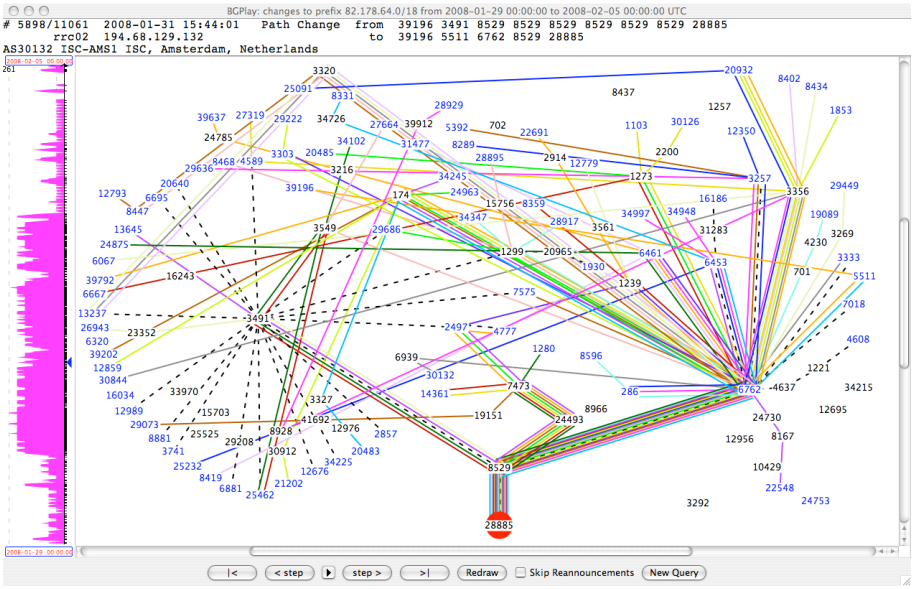


Figure 20: **15:44 (UTC), 31 January 2008** Yet another routing state. AS6762 is used by more peers than ever, AS24493 (Singapore) is still strong and AS3491 (PCCW) is the least preferred transit provider.

Conclusions

The case of OmanTel shows how a combination of (likely) routing policy and an explosion in BGP activity increase the routing topology entropy for AS28885. The number of observed distinct AS paths for the prefixes announced from Oman doubled and the average AS path length increased by 20%. Because there was a constant high rate of changes, we ca not be sure if BGP ever converged in that period, or if the routes which were seen could really be used.

Bibliography

[ACK+08] A. Antony, L. Cittadini, D. Karrenberg, R. Kisteleki, T. Refice, T. Vest, and R. Wilhelm. Mediterranean Fiber Cable Cut (January-February 2008) Analysis of Network Dynamics. Technical Report RT-DIA-124-2008, Dept. of Computer Science and Automation, University of Roma Tre, 2008.

[AKK+08] A. Antony, D. Karrenberg, R. Kisteleki, T. Refice, and R. Wilhelm. YouTube Hijacking (February 24th 2008) Analysis of BGP Routing Dynamics. Technical Report RT-DIA-123-2008, Dept. of Computer Science and Automation, University of Roma Tre, 2008.

[AVG+99] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). IETF RFC 2622, 1999.

[BDPR05] L. Blunk, J. Damas, F. Parent, and A. Robachevsky. Routing Policy Specification Language next generation (RPSLng). IETF RFC 4012, 2005.

[bgpa] BGPath. `http://nero.dia.uniroma3.it/rca/`.

[bgpb] BGPlay RIS. http://www.ris.ripe.net/bgplay/.

[bgpc] BGPlay RV. http://bgplay.routeviews.org/bgplay/.

[cai] CAIDA's AS ranking. `http://as-rank.caida.org/`.

[CBD+02] A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the High Level Structure of the

Internet with Hermes. *Journal of Graph Algorithms and Applications*, 2002.

[CCD+08] A. Campisano, L. Cittadini, G. Di Battista, T. Refice, and C. Sasso. Tracking Back the Root Cause of a Path Change in Interdomain Routing. In *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2008.

[CGH03] D. Chang, R. Govindan, and J. Heidemann. The Temporal and Topological Characteristics of BGP Path Changes. In *ICNP*, 2003.

[Che00] E. Chen. Route Refresh Capability for BGP-4. IETF RFC 2918, 2000.

[com] ROMA TRE Compunet Research Group. `http://www.dia.uniroma3.it/~compunet/`.

[CRC+08] L. Cittadini, T. Refice, A. Campisano, G. Di Battista, and C. Sasso. Measuring and Visualizing Interdomain Routing Dynamics with BGPath. In *IEEE Symposium on Computers and Communications (ISCC)*, 2008.

[CSK03] M. Caesar, L. Subramanian, and R. H. Katz. Towards Localizing Root Causes of BGP Dynamics. Technical Report UCB/CSD-04-1302, EECS Department, University of California, Berkeley, 2003.

[DEH+07] G. Di Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, and T. Schank. Computing the Types of the Relationships between Autonomous Systems. *IEEE/ACM Transactions on Networking*, 2007.

[DKF+07] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley. AS Relationships: Inference and Validation. *ACM SIGCOMM Computer Communication Review*, 2007.

[DKR05] X. A. Dimitropoulos, D. V. Krioukov, and G.F. Riley. Revisiting Internet AS-Level Topology Discovery. In *PAM*, 2005.

[DRCD09] A. Di Menna, T. Refice, L. Cittadini, and G. Di Battista. Measuring Route Diversity in the Internet from Remote Vantage Points. In *International Conference on Networks (ICN)*, 2009.

[DRR06] G. Di Battista, T. Refice, and M. Rimondini. How to Extract BGP Peering Information from the Internet Routing Registry. In *ACM SIGCOMM MineNet Workshop*, 2006.

[FMM+04] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet Routing Instabilities. In *ACM SIGCOMM*, 2004.

[Gao01] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 2001.

[HFLX07] Y. Huang, N. Feamster, A. Lakhina, and J. Xu. Detecting Network Disruptions with Network-Wide Analysis . In *ACM SIGMETRICS*, 2007.

[irl] Internet Topology Project. `http://irl.cs.ucla.edu/topology/`.

[irra] Overview of the IRR. `http://www.irr.net/docs/overview.html`.

[irrb] Internet Routing Registry Daemon (IRRd). `http://www.irrd.net/`.

[irrc] Internet Routing Registry Toolset (IRRToolSet). `http://www.isc.org/index.pl?/sw/IRRToolSet/`.

[Ker02] S. Kerr. RIPE Database Inconsistencies. RIPE Meeting 43, 2002.

[Kon03] H. Kong. The Consistency Verification of Zebra BGP Data Collection. Technical report, Agilent Labs, 2003.

[LCD04a] A. Lakhina, M. Crovella, and C. Diot. Characterization of Network-Wide Anomalies in Traffic Flows. In *IMC*, 2004.

[LCD04b] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-Wide Traffic Anomalies. In *ACM SIGCOMM*, 2004.

[LCD05] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *SIGCOMM Computer Communication Review*, 2005.

[LMJ99] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of internet routing instability. In *IEEE INFOCOM*, 1999.

[LMZ04] M. Lad, D. Massey, and L. Zhang. Link-Rank: A Graphical Tool for Capturing BGP Routing Dynamics. In *IEEE/IPIF NOMS*, 2004.

[LNMZ04] M. Lad, A. Nanavati, D. Massey, and L. Zhang. An Algorithmic Approach to Identifying Link Failures. In *PRDC*, 2004.

[LOMZ07] M. Lad, R. Oliviera, D. Massey, and L. Zhang. Inferring the Origin of Routing Changes using Link Weights. In *ICNP*, 2007.

[LPC+04] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural Analysis of Network Traffic Flows. In *Proc. of ACM SIGMETRICS*, pages 61–72, New York, NY, USA, 2004. ACM Press.

[LWVA01] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja. The Impact of Internet Policy and Topology on Delayed Routing Convergence. In *IEEE INFOCOM*, 2001.

[meda] Mediterranean, Red Sea and Black Sea Region. `http://iscpc.org/cabledb/Mediterranean_and_Red_Sea_Cable_db.htm`.

[medb] Submarine Cable Improvement Group: Worldwide Trends in Submarine Cable System faults. `http://www.scig.net/Section11a.pdf`.

[MFM+06] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-Topology Model that Captures Route Diversity. *ACM SIGCOMM Computer Communication Review*, 2006.

[MKF+06] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat. The Internet AS-Level Topology: Three Data Sources and One Definitive Metric. *ACM SIGCOMM Computer Communication Review*, 2006.

[MSO+99] D. Meyer, J. Schmitz, C. Orange, M. Prior, and C. Alaettinoglu. Using RPSL in Practice. IETF RFC 2650, 1999.

[MUF+07] W. Muhlbauer, S. Uhlig, B. Fu, M. Meulle, and O. Maennel. In Search for an Appropriate Granularity to Model Routing Policies. In *ACM SIGCOMM*, 2007.

[nem] Nemecis. http://ira.cs.ucr.edu:8080/Nemecis/.

[OZP+06] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang. Quantifying Path Exploration in the Internet. In *IMC*, 2006.

[QU05] B. Quoitin and S. Uhlig. Modeling the Routing of an Autonomous System with C-BGP. In *IEEE Network Magazine*, 2005.

[rad] RADB database. ftp://ftp.radb.net/radb/dbase/.

[rega] ARIN Regional Internet Registry. http://www.arin.net/.

[regb] LEVEL3 Local Internet Registry. http://rr.level3.net/.

[regc] RIPE Regional Internet Registry. http://www.ripe.net/.

[regd] VERIO Regional Internet Registry. http://rr.verio.net/.

[ripa] RIPE NCC. http://www.ripe.net.

[ripb] RIPE database. ftp://ftp.ripe.net/ripe/dbase/.

[ripc] The Internet Routing Registry: History and Purpose. http://www.ripe.net/db/irr.html.

[ripd] Mediterranean Fibre Cable Cut - a RIPE NCC Analysis. http://www.ripe.net/projects/reports/2008cable-cut/index.html.

[ripe] RIR delegation statistics. ftp://ftp.ripe.net/pub/stats/ripencc/.

[ripf] YouTube Hijacking: A RIPE NCC RIS case study. http://www.ripe.net/news/study-youtube-hijacking.html.

[risa] RIS search page. http://www.ris.ripe.net/perl-risapp/risearch.html.

[risb] RIS whois. http://riswhois.ripe.net.

[risc] RIS whois Web Interface. http://www.ris.ripe.net/cgi-bin/riswhois.cgi.

[RL95] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). IETF RFC 1771, 1995.

[RLH06] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). IETF RFC 4271, 2006.

[roua] Oregon Route Views Project. `http://www.routeviews.org/`.

[roub] Routing Information System. `http://www.ripe.net/ris/`.

[rrc] Routing Registry Consistency Check (RRCC). `http://www.ripe.net/projects/rrcc/`.

[RSRD07] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for traffic anomaly detection. In *ACM SIGMETRICS*, 2007.

[SF04] G. Siganos and M. Faloutsos. Analyzing BGP Policies: Methodology and Tool. In *IEEE INFOCOM*, March 2004.

[SGK+01] J. Schmitz, E. Gunduz, S. Kerr, A. Robachevsky, and J. L. S. Damas. Routing Registry Consistency Check (RRCC). RIPE Document 201, 2001.

[TR04] R. Teixeira and J. Rexford. A Measurement Framework for Pinpointing Routing Changes. In *ACM SIGCOMM Workshop on Network Troubleshooting*, 2004.

[WMRW05] J. Wu, Z. M. Mao, J. Rexford, and J. Wang. Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network. In *NSDI*, 2005.

[WMW+06] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. *ACM SIGCOMM CCR*, 2006.

[WZP+02] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and Analysis of BGP Behavior Under Stress. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[XCZ05] K. Xu, J. Chandrashekar, and Z. Zhang. A First Step to Understand Inter Domain Routing Dynamics. In *ACM SIGCOMM MineNet Workshop*, 2005.

[ZGGR05] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network Anomography. In *IMC*, 2005.

[ZKL+05] B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang. Identifying BGP Routing Table Transfers. In *ACM SIGCOMM MineNet Workshop*, 2005.

[ZLMZ05] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-Level Topology. *ACM SIGCOMM Computer Communication Review*, 2005.

[ZRF05] J. Zhang, J. Rexford, and J. Feigenbaum. Learning-Based Anomaly Detection in BGP Updates. In *ACM SIGCOMM MineNet Workshop*, 2005.

[ZYZ+04] K. Zhang, A. Yen, X. Zhao, S.F. Wu, D. Massey, and L. Zhang. On Detection of Anomalous Routing Dynamics in BGP. In *Networking*, 2004.

[ZZM+05] X. Zhao, B. Zhang, D. Massey, A. Terzis, and L. Zhang. The Impacts of Link Failure Location on Routing Dynamics: A Formal Analysis. In *ACM SIGCOMM Asia Workshop*, 2005.