Università degli Studi di Roma Tre
**Dipartimento di Informatica e Automazione**
Via della Vasca Navale, 79 – 00146 Roma, Italy

# Active BGP Probing
## Revised Version

L. Colitti[1 2], G. Di Battista[1], M. Patrignani[1], M. Pizzonia[1], and M. Rimondini[1]

**RT-DIA-102-2005** **Novembre 2005**

(1) Dipartimento di Informatica e Automazione,
Università di Roma Tre,
Rome, Italy.
{colitti,gdb,patrigna,pizzonia,rimondin}@dia.uniroma3.it

(2) RIPE – Réseaux IP Européens
Network Coordinating Center,
Singel 258
1016 AB Amsterdam, The Netherlands.
lorenzo@ripe.net

*Work partially supported by European Commission – Fet Open project DELIS – Dynamically Evolving Large Scale Information Systems – Contract no 001907, by MIUR under Project ALGO-NEXT (Algorithms for the Next Generation Internet and Web: Methodologies, Design, and Experiments), and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR Fondo per gli Investimenti della Ricerca di Base.*

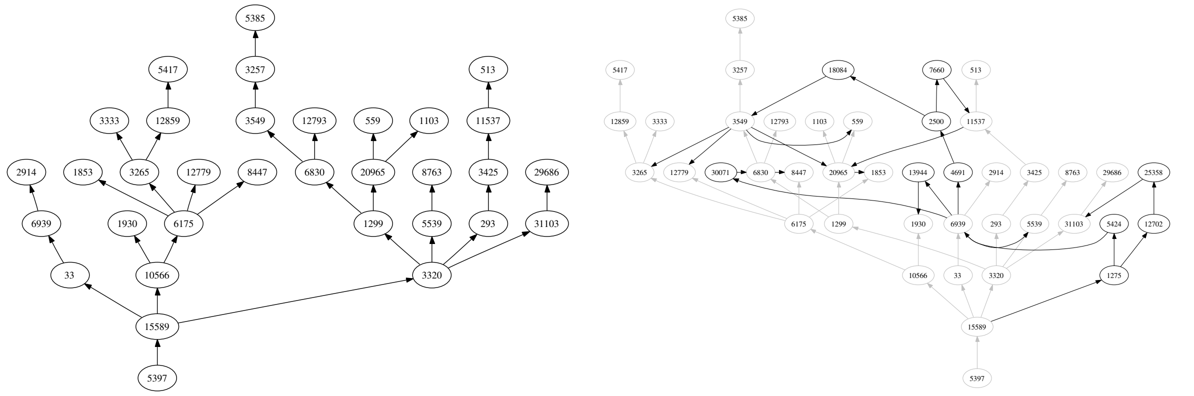
## ABSTRACT

For an Internet Service provider (ISP), the knowledge of which interdomain paths are traversed by its BGP announcements – and thus traffic flows – is essential to predict the impact of network faults, to perform effective traffic engineering, to develop peering strategies, and to assess the quality of connectivity provided by the ISP's upstreams. We present methodologies to discover how the BGP announcements for an ISP's prefix are propagated in the Internet, overcoming the limitations of passive observation of BGP routing tables by actively probing the network using specific BGP updates. The techniques do not require any changes to current operational practices or BGP implementations. We also show how our techniques may be used to determine the routing policies of other ISPs with respect to the ISP's prefix. We validate our techniques through experimentation in the IPv6 Internet, discuss their possible application to IPv4, and compare their results to more traditional topology discovery techniques. We also discuss the operational impact of our techniques and possible ethical concerns arising from their use.

# 1 Introduction

An Internet Service Provider which operates an Autonomous System (AS) $Z$ has a very limited knowledge of the portion of the Internet surrounding $Z$, and this knowledge dramatically decreases with the distance from $Z$. Clearly, the ISP knows its upstream providers. However, it typically has only partial knowledge of their upstream providers, and at AS-distance three the knowledge of how the network treats $Z$'s prefixes becomes negligible.

Consider, as an example, the drawing in Fig. 1. Fig. 1(a) shows what an operator of AS 5397 may discover about the routing of its IPv6 prefix 2001:a30::/32 on Dec 30 2004 at 02:44:00 UTC by querying the RIPE NCC RIS service [20]. An arrow from an AS $A$ to an AS $B$ in Fig. 1(a) means that an announcement of the specified prefix was made by $A$ to $B$. This can be inferred because an AS-path containing the pair $B$ $A$ was registered by a RIS route collector. The graph shows that AS 5397's upstream provider (AS 15589) propagates the announcement to AS 10566, AS 3320 and AS 33, but cannot see, for example, that AS 15589 is propagating the announcement to AS 1275 as well.




(a) (b)

Figure 1: (a) What an operator of AS 5397 may discover about the routing of its prefix 2001:a30::/32 on Dec 30 2004 at 02:44:00 UTC using standard routing information services. (b) Additional topology discovered by sending one custom BGP announcement which includes ASes 33, 3320 and 10566 in an AS-set.

This lack of information has several disadvantages. First, the ISP does not have a clear perception of what will happen to routing in the case of network faults. Second, the lack of in-depth information on how the prefixes announced by the ISP are propagated to the rest of the Internet hampers effective traffic engineering. Third, it is difficult to assess the richness of connectivity and backup paths provided by the ISP's upstreams.

Fig. 1(b) shows the additional information that may be obtained by sending out a single BGP announcement containing a custom AS-set, a technique which we name *AS-set stuffing* and which we discuss in Section 3. Fig. 2 shows the result of a more complete exploration performed using the techniques described in this paper, which reveals a larger set of upstreams for AS 15589 and the complexity of the Internet behind them. The increase in ASes and peerings traversed by the prefix is immediately apparent.

Much has been written on the subject of interdomain topology discovery in the Internet. For example, the behavior of BGP has been passively observed by capturing and

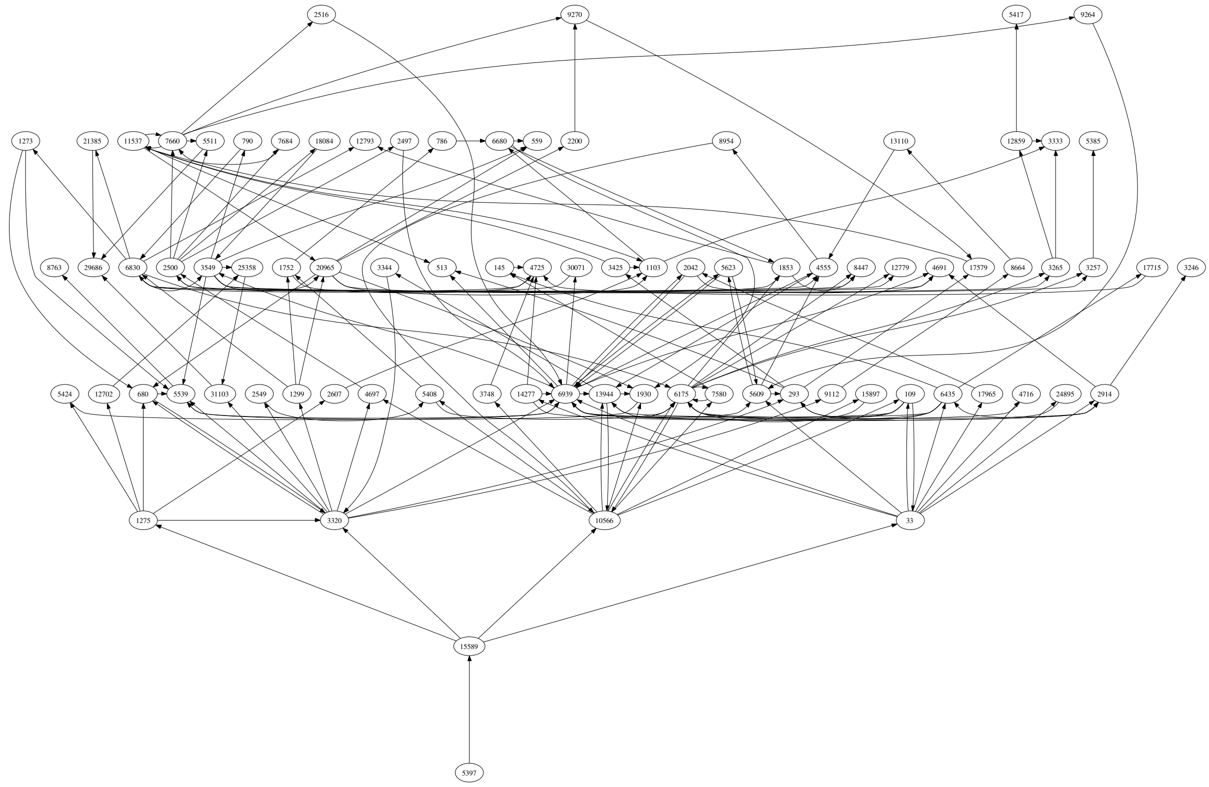


Figure 2: What an operator can see using the techniques described in this paper, in the same situation as in Fig. 1.

analyzing BGP routing tables [3] and announcements [22], and various techniques have been proposed to deduce the interdomain structure using probe packets [9, 24, 16, 7, 17]. Recently, BGP beacons [14] have been used to actively probe the network, with the main purpose of studying network dynamics.

Recent research activity aimed at augmenting the knowledge of interdomain topology through the passive observation of BGP dynamics [28, 4], although more effective than previous techniques in that it discovers hidden links, does not provide information on how a specific prefix is seen by the Internet and how it might be seen by the rest of the network in the event of link faults, changes in routing, or different traffic engineering strategies. While our techniques also observe BGP routing dynamics, they allow an ISP to obtain this information by actively manipulating the BGP announcements for the specific prefix of interest; furthermore, since they alter the Internet routing to a prefix in a stable way, the observation of hidden links does not require the availability of BGP updates but may be performed by querying any looking glass on the Internet, thus greatly increasing the number of views that may be employed for topology discovery.

The contributions of this paper are as follows:

- We present active probing primitives intended to influence how the announcements of a given prefix propagate through the Internet. The primitives rely on standard BGP.
- We exploit these primitives to devise new discovery algorithms. Namely, we present

algorithms that, given a prefix $p$ originated by a certain AS $Z$, can be used to discover which ASes and peerings the BGP announcements of $p$ can traverse, with special attention to the portion of the Internet surrounding $Z$. We also show how our probing primitives can be used to check, at least partially, the policies of a given AS $A$ with regard to a certain prefix $p$. Namely, we show how to use our primitives: (i) to check if a certain AS path through $A$ is permitted by the current routing policies, (ii) to infer, given two AS paths through $A$, which one is preferred.

- In order to determine the degree of feasibility of our techniques and ensure that they do not cause any problems to the stability of the Internet, we evaluate their impact on equipment and routing practices in use in today's network.

- We extensively test our techniques on the IPv6 and IPv4 Internet and show their effectiveness.

The rest of the paper is organized as follows: in Section 2 we provide background information. In Section 3 we introduce our primitives, and in Section 4 we present their applications. In Section 5 we discuss the operational impact and the possible ethical issues posed by our techniques. Experimental results are presented in Section 6, and we conclude in Section 7.

# 2 Background

An Internet Service Provider (ISP) typically administers one or more *Autonomous Systems* (ASes). An AS is a portion of the Internet under a single administrative authority and is identified by an integer number. ASes exchange routing information with other ASes by means of a routing protocol called Border Gateway Protocol (*BGP*) [19, 26]. Two ASes that directly exchange routing information are said to have a *peering* between them. The ASes that have peerings with an AS $A$ are termed the *peers* of $A$.

BGP operates on blocks of contiguous IP addresses known as *prefixes*. The sequence of ASes traversed by traffic sent to a particular prefix is determined by the *AS-path* attribute associated with the prefix. A $\langle \text{prefix}, \text{AS-path} \rangle$ pair is known as a *route*. BGP peers exchange routes using BGP *update* messages. A BGP update is either a route *announcement* or a route *withdrawal*. An announcement conveys the following information: "through me you can reach a certain prefix; to reach it, I will use the following AS-path". A withdrawal nullifies a previously communicated route for a specified prefix. In other words a withdrawal means "you can no longer reach this prefix through me". The BGP specification [19] states that AS-paths may be composed of an arbitrary number of AS-set or AS-sequence elements. The *AS-sequence* is an ordered list of ASes, while the *AS-set* is an unordered set of ASes. In practice, the vast majority of BGP announcements are composed of a single AS-sequence, possibly followed by an AS-set in certain cases of route aggregation.

Each router stores information on how to reach every prefix it is aware of in its *routing information base* (*RIB*), which is a table of routes. For each prefix in the RIB, one of the routes to it is chosen to be the *best* and used in IP forwarding.

Routes related to a certain prefix begin their existence within an AS called the *originator* of the prefix (typically the AS to which the prefix belongs). The routes are propagated by means of route announcements to peer ASes. A router which receives an announcement inserts the route into the RIB and recalculates the best route to the prefix. If the best route has changed, it propagates the new best route to its peers. Every time a router propagates an announcement to another AS, it prepends its AS identifier to the AS-path it received; thus, the AS-path of an announcement is generally the list of ASes that the announcement has passed through. In order to avoid routing loops, if a BGP router receives an announcement which includes the number of its own AS in the AS-path, it discards the announcement without further processing.

A prefix $p$ is selectively propagated by an AS $A$ to its peers depending on the routing policy adopted by $A$ and by the contents of the announcement [6, 10, 11]. We say that an AS-path $A_n \dots A_2 A_1$, where $A_1$ is the origin AS, is *feasible* for a prefix $p$ if the policies of each $A_i$ permit $A_i$ to announce $p$ to $A_{i+1}$ with AS-path $A_i \dots A_1$. Observe that the feasibility of an AS-path for a prefix does not imply that the AS-path is necessarily visible in the Internet: it only means that under certain circumstances it may be visible. Thus, the set of feasible AS-paths for a prefix $p$ contains all the AS-paths that may be observed for $p$ in the Internet. We note that the concept of feasible path has also been used in the literature on the stability of BGP with the name of *permitted* path (see e.g. [8]). A peering between two ASes $P$ and $Q$ is feasible for $p$ in the direction from $P$ to $Q$ if there exists at least one feasible path $A_n \dots QP \dots A_1$ (where it may be that $Q = A_n$ and/or $P = A_1$). In diagrams we shall represent a feasible peering from $P$ to $Q$ with a directed arc from $P$ to $Q$. For example, in Fig. 1(a) the directed arc between 10566 and 6175 indicates that the policies of AS 10566 permit it to announce the prefix 2001:a30::/32 to AS 6175.

We name *routing state* for a prefix $p$ at a given time the set of best routes to $p$ of each router in the Internet at that time. We empirically say that a routing state for $p$ is *stable* if we have observed no BGP updates for $p$ for a sufficiently large time interval.

To obtain (partial) information about the evolution of the Internet routing state, projects such as the RIPE NCC Routing Information Service (RIS) [20] or the University of Oregon's RouteViews Project [18], deploy *route collectors* in specific points of the Internet to record BGP updates from a number of ASes, which we name *collector-peers*. Thus, the best routes of each collector-peer at any given time are known. The RIB of the collectors and the updates they receive are periodically dumped, permanently stored and made publicly available over the Web.

# 3 BGP Probing Primitives

Consider the case of an AS $Z$ that announces a prefix $p$. In this section we present basic primitives for active BGP probing which can be used by the operators of $Z$ to obtain information on how $p$ is propagated in the Internet. The primitives are based on sending BGP updates for $p$ and observing the effects, which may lead to the discovery of alternate paths (and, from these, ASes and peerings), that are feasible but are not ordinarily used (e.g., backup paths).

## 3.1 AS-set Stuffing

In the following we denote an announcement for $p$ with AS-path $\mathcal{P}$ as $\langle p, \mathcal{P} \rangle$. The announcement for $p$ sent by $Z$ to its peers is usually $\langle p, Z \rangle$. To prevent announcements for $p$ from traversing an arbitrarily chosen AS $A$, $Z$ may announce $\langle p, ZA \rangle$. Since a BGP router discards any announcement whose AS-path contains its own AS number, $A$ will discard the announcement and will not propagate it to any other AS. The same approach can be used to exclude any set of ASes $A_1, A_2, \ldots, A_n$ by announcing $\langle p, ZA_1A_2 \ldots A_n \rangle$. We name these ASes *prohibited* ASes.

With this approach, the number of prohibited ASes influences the length of the AS-path, which is not desirable since the AS-path length is one of the most important metrics used by BGP routers in the route selection process. To avoid this, the prohibited ASes are put into an AS-set at the end of the AS-path.

Our first primitive, which we name *AS-set stuffing*, consists in announcing $p$ with an AS-path of $Z\{A_1A_2 \ldots A_n\}$, where $\{A_1A_2 \ldots A_n\}$ is an AS-set. The length of the resulting AS-path is counted as two, since the length of an AS-set is typically considered to be one irrespective of the number of ASes in it.

The observation of the resulting routing state, and possibly of the convergence process, allows us to determine alternative feasible paths for $p$ that do not contain the prohibited ASes.

## 3.2 Withdrawal Observation

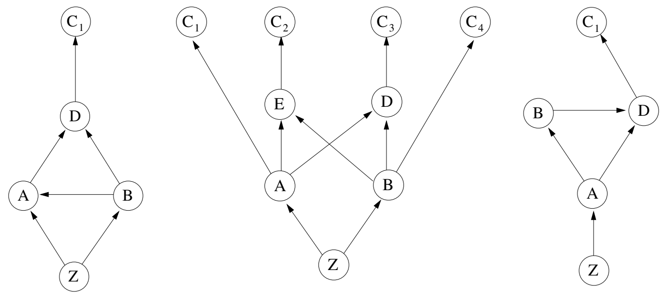
When the originator of a particular route withdraws the route, the withdrawal does not immediately reach all the ASes in the network; instead, it propagates across the network in a potentially lengthy (usually lasting several minutes) convergence process which generates a large number of BGP updates, as ASes which do not yet know that the route has been withdrawn switch to alternate paths. This is because each AS which receives a withdrawal only sends out a withdrawal itself if it knows no other routes to the destination; otherwise, it will simply send out an update message with one of the alternate paths it knows [13].

Our second primitive, which we name *withdrawal observation*, consists in sending a withdrawal for $p$ and observing all the paths that become visible during the BGP convergence process.

Although withdrawal observation does not offer the control and flexibility of AS-set stuffing, it has the advantage that it is quick and easy to perform: all that is needed is to send a withdrawal and observe the results.

## 3.3 Limitations and Constraints

Our primitives allow the observation of feasible peerings for a given prefix $p$ which are not ordinarily visible in the absence of active probing. However, there are intrinsic limitations to what we may observe. Consider, for example, Fig. 3(a). If $Z$ is the originator of $p$ and $C_1$ is a collector, the use of AS-set stuffing does not guarantee that it is possible to observe the peering between $B$ and $A$, although it is feasible. The peering cannot be observed in a stable routing state, because the only probes that can be performed are to prohibit $A$ and/or $B$, but in every case the peering will not be visible since one of its endpoints is prohibited. However, a path such as $C_1DABZ$ may be visible during BGP convergence, depending on the unpredictable order of updates propagated in the network. In this case, our primitives will observe the peering.




(a) (b) (c)

Figure 3: $Z$ is the origin AS, while $C_1$, $C_2$, $C_3$, and $C_4$ are collectors. An edge directed from $x$ to $y$ represents a peering that is feasible in that direction. (a) A topology where AS-set stuffing does not guarantee the discovery of the feasible peering between $B$ and $A$. (b) A topology in which level-by-level exploration cannot discover all feasible peerings. (c) A topology in which the feasibility of the path $ZABDC_1$ cannot be determined.

Furthermore, the number of ASes that may appear in an AS-set is limited: first, the length of an AS-set is specified by a 1-byte field in the BGP packet, which poses a limit of 255 ASes. Second, as we shall see in Section 5.1, the BGP implementations of many commercial routers pose other limits to the number of ASes that may appear in an AS-path. As we show in Section 6, this is unlikely to be an issue in the application of our techniques.

A constraint of a different nature is posed by route flap dampening [15], which limits the propagation of frequent updates for the same prefix, thus imposing rate-limiting on our active BGP probing. The exact effects of route flap dampening depend on the topology, but [15] suggests that the maximum length of time a route can be suppressed by dampening in today's Internet is approximately one hour. The degree to which route flap dampening impacts our discovery strategies is discussed separately for each technique in Section 4. Experimental results on the effect of dampening are presented in Section 6.

Finally, we note that the portion of the Internet that can be observed is limited by the number of views provided by the route collectors used. This is true of all topology discovery algorithms that use route collectors. However, unlike techniques that rely on observing BGP updates to discover hidden ASes and peerings, the use of AS-set stuffing allows us to alter the routing so that alternate paths are used in stable routing states. Therefore, observation of ASes and peerings that are not ordinarily visible does not require access to BGP routing updates, but may be performed using (possibly automated) queries to any accessible looking glass on the Internet, thus greatly increasing the number of views that may be employed in topology discovery.

# 4 Discovery Strategies

In this section, we present several applications of the primitives introduced in Section 3. We again consider the case of an AS $Z$ that announces a prefix $p$. First, we propose methods to discover peerings in the vicinity of $Z$ that are feasible for $p$. Second, we show

how $Z$ can check if a certain AS-path originating from $Z$ is feasible for $p$. Finally, we show how, given two feasible AS-paths starting from $Z$ and ending in an AS $A$, $Z$ can determine which one is preferred by $A$. Our general approach can be summarized as follows:

(i) Originate a suitable BGP update from $Z$;

(ii) Wait for the network to reach a stable state;

(iii) Retrieve the updates recorded during the BGP convergence process and the final routing state from the route collectors;

(iv) Infer how the network behaves with respect to $p$.

The process can be iterated several times.

## 4.1 Prefix Propagation Discovery

We name *complete feasibility graph* for a prefix $p$ the directed graph whose nodes are ASes and whose arcs are feasible peerings for $p$. Observe that every feasible AS-path is a directed path in the complete feasibility graph. The reverse is not generally true because of routing policies: for example, an AS $B$ may decide to propagate the announcement for $p$ to an adjacent AS $C$ only if it receives it from AS $D$ and not if it receives it from AS $E$. Thus, any AS-path containing the sequence $E \rightarrow B \rightarrow C$ is not feasible for $p$, even though $E \rightarrow B$ and $B \rightarrow C$ are feasible peerings.

Ideally, $Z$ would like to know the complete feasibility graph for $p$. However, this would require the knowledge of the routing policies of every BGP router in the Internet, which is not realistic. Therefore, we introduce the concept of feasibility graph: consider a certain exploration process undertaken by $Z$ to discover the feasible peerings for $p$. We name *feasibility graph* the subgraph of the complete feasibility graph which has so far been discovered in the course of the exploration.

In the following, we name *level* of a node $X$ in the feasibility graph the length of the shortest directed path from $Z$ to $X$. Note that the feasibility graph changes over time due to topology and configuration changes, but in the following we assume that it does not change in the course of a specific discovery process.

AS $Z$ may obtain a feasibility graph at a given time using a query to the route collectors, but the extent of such a graph is limited, as can be seen in Fig. 1. Our strategies for prefix propagation discovery permit $Z$ to obtain a much larger feasibility graph.

Both of the primitives described in Section 3 can be used for this purpose. $Z$ may use withdrawal observation: for every AS-path that is announced during the BGP convergence process, discover the corresponding nodes and arcs of the feasibility graph. However, the portion of the graph that is discovered is not under the direct control of $Z$, but depends on the order in which BGP updates are propagated.

The use of AS-set stuffing gives $Z$ much greater flexibility in the discovery process. In principle, $Z$ could send $2^n$ announcements, including in each one an AS-set containing one of the $2^n$ subsets of the $n$ ASes in the Internet. This would provide the most complete knowledge of the feasibility graph possible with AS-set stuffing. Obviously, such a brute force approach is infeasible both because the number of ASes that may be included in an AS-set is limited and because of the long exploration times it would require. Therefore, we adopt the following strategy: begin with the directed AS graph seen by the route collectors at a certain instant and proceed level by level, starting from level one. For each level, prohibit all the known ASes in the level. At this point, either there will be no feasible paths to the collectors, or the announcements will propagate through new, previously unknown, nodes at the same level. Each new node and arc found is added to the feasibility graph. If new nodes in the same level have been found, insert them into the prohibited set; otherwise, empty the set of prohibited ASes and proceed to the next level. As an example, Fig. 1(b) shows the new nodes and arcs discovered starting from the situation in Fig. 1(a) by announcing the AS-set $\{33, 3320, 10566\}$, which corresponds to all the known nodes at level two in the initial graph.

After every BGP update, we wait a period of time to allow the network to converge and to limit the effects of route flap dampening. To deduce the presence of nodes and links that might not be visible in stable states, we examine all the updates received for $p$ during the convergence period. A more formal description of the algorithm follows. We denote with `F` the feasibility graph, which is initially empty and is incrementally constructed during the execution of the algorithm, and with `F[lev]` the nodes of level `lev` in `F`. We use the same notation for a temporary graph `G`. We denote with `p` the prefix announced by $Z$. We name this algorithm the *level-by-level* exploration algorithm.

```
### Level-by-level exploration algorithm

# Obtain initial graph
F = query_route_collectors(p)

# Explore one level at a time
lev = 1
while F[lev] not empty:

    # Progressively prohibit ASes until
    # no new nodes are found in level lev
    newnodes = F[lev]
    while newnodes not empty:

        # Announce AS-set containing all
        # known nodes at level lev
        announce_as_set(F[lev])

        wait_for_bgp_convergence()

        # Query route collectors, merge
        # new nodes and arcs found into F
        previous = F[lev]
        G = query_route_collectors(p)
        F = merge_graphs(F, G)
        newnodes = F[lev] - previous

    lev = lev + 1
```

We note that this algorithm, in addition to the intrinsic constraints of AS-set stuffing described in Section 3.3, suffers from further limitations. For example, in Fig 3(b), the algorithm does not guarantee that all the arcs $A \to D$, $B \to E$, $A \to E$, and $B \to D$ will be discovered.

Another possible algorithm based on the same primitive is as follows: once all nodes in a level $l$ have been found, process each node in $l$ in turn. For each one, prohibit all the other nodes in $l$, then progressively prohibit all visible nodes in level $l+1$ until no new nodes in level $l+1$ are found. Then empty the set of the prohibited nodes and advance to the next node in $l$. We name this algorithm the *node-by-node* exploration algorithm. Node-by-node exploration overcomes the limitations of level-by-level exploration; however, it requires many more updates, and therefore, due to the effect of route flap dampening, longer exploration times. We believe that level-by-level exploration is a good compromise between speed and completeness of results.

Furthermore, we extract data from all the route collectors simultaneously; however, in certain topologies, using only one collector at a time and merging the results at the end allows the discovery of more arcs. For example, in Fig. 3(b), exploring the topology separately using level-by-level exploration, first using only $C_2$ and then only $C_3$, would also reveal the arcs $B \to E$ and $A \to D$, while exploring the topology using both collectors simultaneously might only discover the arcs $A \to E$ and $B \to D$.

The development of more sophisticated topology discovery algorithms based on AS-set stuffing is an area we leave to further research.

## 4.2 Path Feasibility Determination

Suppose that we are interested in knowing whether a certain AS-path $\mathcal{P}$ is feasible for a prefix $p$. For this purpose, we may use the following algorithm. Suppose that $\mathcal{P}$ ends at a collector-peer $C$, and let $\mathcal{S}$ be the (initially empty) AS-set that is used in the announcements of prefix $p$. Also let $|\mathcal{P}|$ be the length of path $\mathcal{P}$. Repeat the following steps:

– Announce prefix $p$ with AS-set $\mathcal{S}$ in the path

– Let $\mathcal{Q}$ be the AS-path seen by collector $C$ for prefix $p$

– Remove consecutive occurrences of the same AS from $\mathcal{Q}$ and $\mathcal{P}$

– If $\mathcal{Q} = \mathcal{P}$, return *feasible*

– If $\mathcal{Q}$ is empty, return *not feasible*

– Compare the ASes in $\mathcal{P}$ and $\mathcal{Q}$ in order, starting from the origin AS, and let $i$ be the position where the two paths differ.

– If $\mathcal{Q}[i]$ appears in $\mathcal{P}$, return *unknown*

– $\mathcal{S} = \mathcal{S} \cup \{Q[i]\}$

For example, in the topology in Fig. 3(c), it is not possible to determine whether the AS-path $ZABDC_1$ is feasible (i.e., the algorithm returns *unknown*). This is due to the intrinsic limitation of AS-set stuffing described in Section 3.3: unless the link between $A$

and $D$ fails, $C_1$ will receive the shortest path $ZAD$, and prohibiting $D$ would eliminate the path whose feasibility we wish to determine. Intuitively, the peering $A \to D$ is a "short cut" which, in the absence of routing policies to the contrary, is always preferred by BGP.

If the path does not end at a collector-peer, consider one collector-peer $C$ and apply the above algorithm, replacing the exit condition $\mathcal{Q} = \mathcal{P}$ with "$\mathcal{P}$ matches the first $|\mathcal{P}|$ AS numbers of $\mathcal{Q}$ starting from the origin AS". In this case, the algorithm returns *unknown* also for the case in which $\mathcal{Q}$ is empty, since it cannot be determined whether $\mathcal{P}$ is unfeasible or the subpath from the end of $\mathcal{P}$ to $C$ is. To obtain more complete results, repeat this process for every collector-peer $C$ in turn until either the exit condition is true or all collectors have been tried.

This algorithm may require up to $N$ different announcements per route collector tried, where $N$ is the number of ASes. Therefore, due to the effect of route flap dampening, it requires very long execution times. A more practical approach, which usually requires only one announcement to obtain the same results and therefore is much less affected by route flap dampening, is the following, which we name the *nailed-path* algorithm: consider the feasibility graph obtained using any of the methods defined in Section 4.1 and prohibit all the ASes in levels up to and including the level of $C$ except for the ASes in $\mathcal{P}$. Now observe the AS-path $\mathcal{Q}$ seen by $C$: it is likely that either $\mathcal{Q} = \mathcal{P}$ and the path is feasible, or $C$ does not see the prefix and the path is not feasible. If $\mathcal{Q} \neq \mathcal{P}$, we execute the above algorithm starting from step (ii). This may happen if the update sent by the algorithm reveals ASes or peerings that were not in the initial feasibility graph, or in one of the cases discussed in Section 3.3. As previously stated, in the latter case the feasibility of $\mathcal{P}$ cannot be determined.

If $\mathcal{P}$ ends in an AS $A$ that is not a collector-peer, we proceed in the same manner, prohibiting all ASes in levels up to and including the level of $A$, and check whether $\mathcal{P}$ is a subpath of one of the paths seen by the route collectors.

## 4.3 Path Preference Comparison

Given two feasible AS-paths $\mathcal{P}_1$ and $\mathcal{P}_2$ ending in the same AS $A$, we may use AS-set stuffing to determine which of the two AS-paths is preferred by $A$. Note that if $\mathcal{P}_1$ and $\mathcal{P}_2$ have ASes in common in addition to $A$, which AS-path $A$ would prefer is irrelevant since $A$ will only ever receive one of the two paths. Therefore, we limit our attention to the case in which $\mathcal{P}_1 \cap \mathcal{P}_2 = \{A, Z\}$.

To determine which path $A$ prefers, we obtain a feasibility graph as described in Section 4.1 and attempt to ensure that the only announcements received by $A$ for $p$ have the paths $\mathcal{P}_1$ and $\mathcal{P}_2$. If $A$ is a collector-peer, we prohibit all the ASes in all levels up to the level of $A$ except the ASes in $\mathcal{P}_1 \cup \mathcal{P}_2$. Usually, this is enough for $A$ to see either $\mathcal{P}_1$ or $\mathcal{P}_2$. However, it may also lead to the discovery of new ASes which are not in $\mathcal{P}_1$ or in $\mathcal{P}_2$ and were not previously visible in the feasibility graph. In this case, it is sufficient to prohibit the new ASes and repeat the announcement until no new nodes are found. The announcement may also lead to the observation of another path made up exclusively of ASes which belong to either $\mathcal{P}_1$ or $\mathcal{P}_2$; in this case it is not possible to determine whether $A$ prefers $\mathcal{P}_1$ or $\mathcal{P}_2$. We note that this may only occur if there is a feasible peering between an AS in $\mathcal{P}_1$ and an AS in $\mathcal{P}_2$. Finally, since this technique requires us to determine whether $\mathcal{P}_1$ and $\mathcal{P}_2$ are feasible, it cannot be applied if it is not possible to determine the feasibility of the paths as described in Section 4.2.

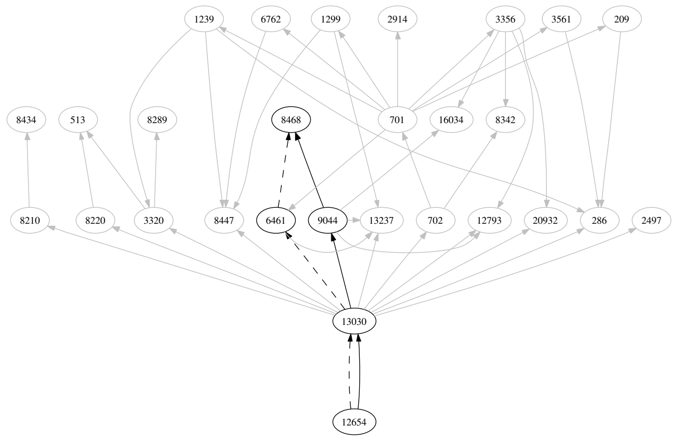



Figure 4: Path preference comparison example: the paths to be compared are 8468 6461 13030 12654 (dashed) and 8468 9044 13030 12654 (solid). The ASes in gray are prohibited.

As an example, consider Fig. 4, where we aim to determine whether AS 8468, a collector-peer, prefers the path through AS 6461 or AS 9044. The graph was obtained using withdrawal observation. Since all the ASes not part of the two paths (shown in gray) are prohibited, the preferred path is the one seen by AS 8468. For details of the experiment, see Section 6.6.

If $A$ is not a collector-peer, a possible approach is the following: choose a collector $C$ and determine two paths $\mathcal{P}'_1$ and $\mathcal{P}'_2$, where $\mathcal{P}'_1(\mathcal{P}'_2)$ is the concatenation of $\mathcal{P}_1(\mathcal{P}_2)$ with $\mathcal{P}_C$ and $\mathcal{P}_C$ is any subpath between $A$ and $C$ such that $\mathcal{P}'_1$ and $\mathcal{P}'_2$ are both feasible. Now $\mathcal{P}'_1$ and $\mathcal{P}'_2$ both end in $C$ and we may apply the above algorithm to determine whether $C$ sees $\mathcal{P}'_1$ or $\mathcal{P}'_2$. Since the two AS-paths coincide from $A$ onwards, which path is seen by $C$ depends on which path is preferred by $A$. This may be repeated for different collector-peers $C$ if necessary.

The effect of route flap dampening on path preference comparison is limited. If we start from a situation in which $p$ is not announced, the number of updates propagated in the network is very low. When verifying the feasibility of $\mathcal{P}_1$ and $\mathcal{P}_2$, the only ASes that process BGP updates are those in the path being tested (plus any previously unknown ASes that are not prohibited by the nailed-path algorithm), and once the feasibility of $\mathcal{P}_2$ has been successfully tested, comparing the paths only requires removing the ASes in $\mathcal{P}_1$ from the prohibited set, which is likely to cause very few updates.

# 5 Operational Impact and Ethical Issues

In this section we discuss the technical limitations and the effects on the Internet of our techniques. We also argue that our techniques, if properly used, are safe and we address ethical issues they might pose.

## 5.1 Feasibility and Impact of Large AS-sets

Although the BGP updates generated by our techniques conform to the BGP specification and should be processed correctly by any compliant router, the unusual length of the AS-sets generated by AS-set stuffing may raise concerns about the robustness of routers in handling such updates.

To discover whether our BGP updates might have an operational impact on the Internet, we examined historical BGP data to determine the incidence of AS-sets and their sizes in the normal operation of the Internet. The results show that AS-sets, although rare, are constantly present in today's Internet. For example, data from the RIS route collector RRC03 for the month of February 2005 showed that every RIB dump contained between 500 and 700 entries which originated in an AS-set. The AS-sets observed are usually not very large: the largest AS-set observed during this period contained 15 ASes. However, AS-sets of unusual length have been observed before. For example, an AS-set containing 123 ASes was observed in January 2001 [1], and one containing 124 ASes was observed in June 2002 [27]. We know of no reports of adverse affects on the network caused by these announcements.

To determine how routers treat large AS-sets, and so see whether our AS-sets can effectively be used in topology discovery in the real world, we also performed tests on equipment from popular router manufacturers. Versions 12.2 and 12.0(17)S of Cisco IOS introduced the `bgp maxas-limit` command, which causes the router not to use or propagate announcements whose AS-path contains more than the configured number of ASes. The default value is 75; however, irrespective of the value of the setting, tests on various releases of IOS showed that the routers would reset the BGP session if they received an AS-path more than 512 bytes long, i.e. where the total number of ASes (including both the ASes in the AS-sequence and in the AS-set) is greater than 254. We note that this is a security problem that may be exploited by a malicious network operator to perform a denial of service attack on neighboring ASes. When we reported the issue to Cisco, the vendor promptly replied that the bug would be fixed in a future release of IOS. We also tested with a Juniper M7i router running JUNOS 7.0R1.5, which had no problems receiving and propagating AS-sets containing up to 255 ASes, the maximum length permitted by the BGP packet format.

As the example AS-sets used in our experiments show (see Section 6), such limits do not pose practical problems to our techniques in the topologies we tested, since the number of nodes involved is generally much lower. Further, we must take into account the fact that more ASes are added to the path as it is propagated in the Internet, and therefore our techniques must "leave room" for propagation, although we note that AS-paths longer than 15 ASes are rare in today's Internet [12].

Another issue to consider is the possible increase of BGP traffic caused by active probing. In this respect, the number of BGP updates received from a Tier-1 router is typically higher than 15,000 per hour on average. Since route flap dampening renders our techniques ineffective if more than roughly one update per hour per prefix is sent, the effect of one exploration on the number of BGP updates seen in the Internet is negligible.

Finally, as regards possible impact on router memory, we expect that the amount of memory needed to store an AS-path with a 100-element AS-set is about 200 bytes larger than the amount of memory needed to store the same AS-path without the AS-set. Compared to the several megabytes (or tens of megabytes) of memory used by a full BGP routing table, the effect of such an AS-set is negligible.

### 5.2 Ethical Issues

In this subsection we consider the techniques introduced in this paper from an ethical point of view.

One possible concern regarding our techniques is that they might cause operational problems: an update with a large number of ASes in the AS-set might cause confusion as to which AS actually originated the update, thus hampering debugging; and the presence of an AS number in an AS-path might suggest that that AS was involved in the update even though in fact it was not. However, we note that the conventional use of AS-sets for route aggregation already suffers from both these problems. For example, an AS-path of 1 $\{2, 3, 4\}$ implies that one of AS2, AS3, or AS4 originated the routing information that generated the update, but does not specify which one. Also, since every BGP announcement is tied to a particular prefix, the origin of the announcement can easily be traced by querying the Internet Routing Registries. Finally, the BGP announcements used can be tagged using BGP community attributes [2] to indicate that they make use of AS-set stuffing, although community values are not always propagated by the ASes they traverse.

A second possible concern is that our use of AS-sets is not the use intended by the BGP specification. This is true, but we note that it frequently happens that protocols designed for a certain purpose are used in ways that the original designers did not foresee. Examples are Network Address Translation [5], which uses the TCP and UDP port fields to distinguish between hosts using private addresses, IP-in-IP tunneling [23], where a layer 3 protocol is carried by another layer 3 protocol instead of by a data-link layer protocol, and the use of duplicated TCP acknowledgements to implement fast retransmit and fast recovery congestion control [25].

Finally, it could be argued that placing the AS number of another AS in a BGP update should require permission from the AS in question. There is no technical reason for this; rather, such an action might be perceived as an improper use of an asset of another organization. We believe that this is a matter of policy to be discussed in the appropriate fora. The opinion of the authors is that, since the techniques cause no technical problems, it is only a matter of determining their cost-benefit ratio. Since our techniques cannot reasonably be used if the originating AS needs to obtain permission from all the ASes that are close to it in the Internet topology, we believe that the question is simply one of deciding whether the techniques are useful for the Internet community or not.

## 6 Experimental results

In this section we describe our experimental setup, discuss the effectiveness of our prefix propagation discovery strategies and the limitations posed by route flap dampening. Further, we test our policy determination techniques. Finally, we relate our discovery techniques to more conventional methods.

### 6.1 Experimental setup

Due to the innovative nature of our techniques, we first tested them in the IPv6 Internet, in order to limit the extent of any possible problems they might cause: the IPv6 Internet is

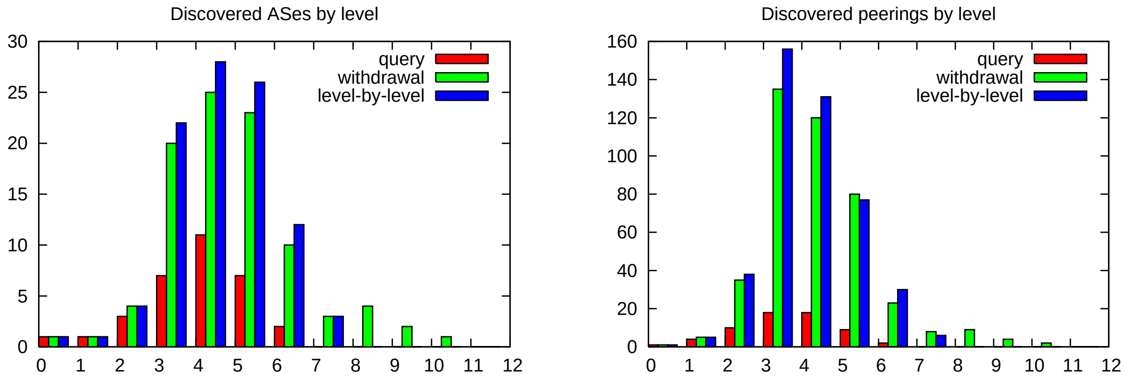



Figure 5: ASes and peerings found by a standard RIS query, by withdrawal observation, and by level-by-level exploration, sorted by level (IPv6).

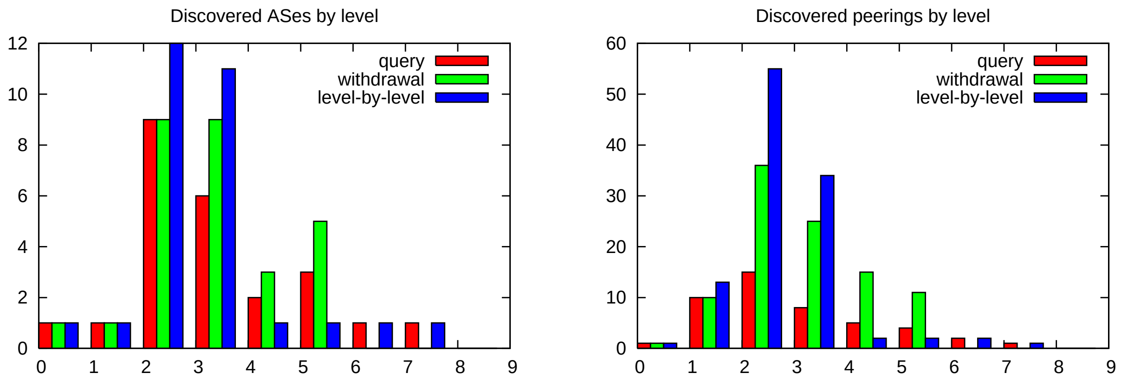



Figure 6: ASes and peerings found by a standard RIS query, by withdrawal observation, and by level-by-level exploration, sorted by level (IPv4).

smaller than the IPv4 Internet, employs fewer legacy devices, and supports fewer mission-critical services. The IPv6 experiments were performed between November 2004 and April 2005. Once we were confident of the reliability and effectiveness of our techniques, we performed more limited testing on the IPv4 Internet in order to verify how the different topologies and operational practices in use affect our techniques. These experiments, with prior warning to various high-profile operational mailing list, were conducted between June 2005 and July 2005.

All BGP updates were generated using custom software developed by the authors [21], and the effect of each BGP update sent was observed by means of the RIS database, which provides BGP data collected in real-time. The IPv4 experiments were not observed using all route collectors due to the temporary unavailability of some collectors in that period.

IPv6 BGP announcements originated in AS 5397 using the prefix `2001:a30::/32`, while IPv4 BGP announcements originated in AS 12654 using the prefixes `84.205.73.0/24` and `84.205.89.0/24`. These prefixes, reserved for BGP experiments, are announced by the RIS route collectors, currently present in 13 locations around the world. In order to approximate the situation of a small to medium-sized ISP, each prefix was announced by one route collector at a time.

Finally, we note that our experiments are difficult to reproduce because they depend

| Method | IPv6 | | IPv4 | |
|---|---|---|---|---|
| | ASes | Peerings | ASes | Peerings |
| Stable state | 32 | 31 | 24 | 23 |
| Withdrawal observation | 94 (2.9x) | 211 (6.8x) | 28 (1.2x) | 49 (2.1x) |
| Level-by-level exploration | 97 (3.0x) | 222 (7.2x) | 29 (1.2x) | 55 (2.4x) |

Table 1: ASes and feasible peerings found by a RIS query in a stable routing state, by withdrawal observation, and by level-by-level exploration.

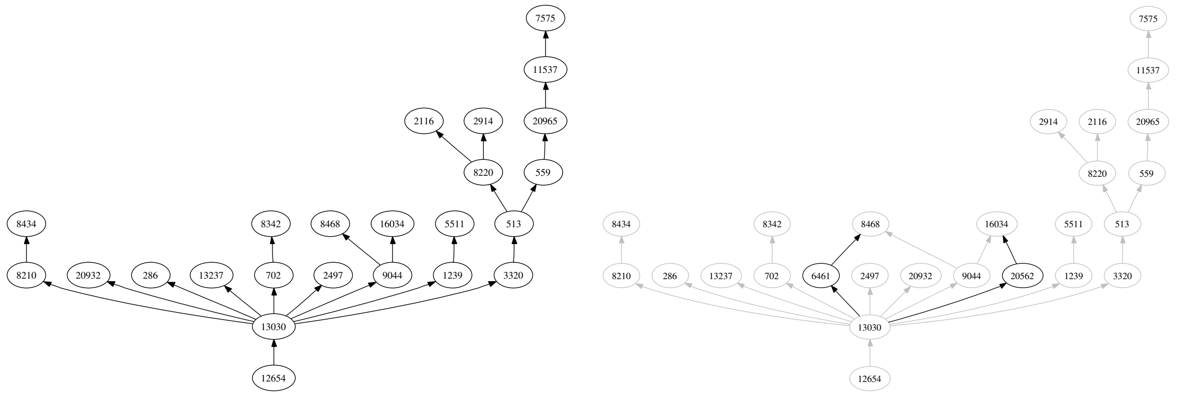
on the current state of the BGP routing in the Internet and the unpredictable order in which BGP updates are propagated; however, the announcements we sent and the results they produced are permanently stored by the RIS in the raw data repository.

## 6.2 Prefix propagation discovery

To evaluate the effectiveness of our topology discovery strategies, we compared the feasibility graphs they generated to graphs obtained from the collectors in stable routing states. The results are in Table 1. As can be seen, in the IPv6 network our techniques observe approximately three times more ASes and seven times more feasible peerings than when active probing is not used. In the IPv4 network our techniques observe approximately 20% more ASes and over twice as many feasible peerings than when active probing is not used. We believe that the difference in effectiveness of our techniques between IPv6 and IPv4 is due to the much more restrictive routing policies employed in the IPv4 Internet.

The results obtained using AS-set stuffing are slightly better than those produced by withdrawal observation. However, although the results are similar in terms of the number of nodes and peerings discovered, the topologies discovered are different. The graphs in Figs. 5 and 6 show the level at which the new ASes and peerings are discovered (we count every peering twice, once at the level of one endpoint and once at the level of the other endpoint). As can be seen from the graph, the topology produced by level-by-level exploration is more concentrated in the lower levels of the feasibility graph. Since the ASes that were discovered by the two methods are mostly the same, this means that certain ASes were discovered at a lower level by level-by-level exploration than by withdrawal observation. Therefore, since the definition of the level of a node is the shortest topological distance in the feasibility graph from the origin AS, the topology produced by level-by-level exploration is more accurate.

Fig. 7 shows an example of how AS-set stuffing may be used to discover the peers of an ISP's upstream provider. Fig. 7(a) shows a stable state feasibility graph for prefix `84.205.73.0/24` announced from RRC11 and observed at 14:49:59 UTC on July 5 2005. The graph shows that AS 13030 is the only upstream of the origin AS (AS 12654) and has 9 visible peers. Fig. 7(b) shows how a single announcement with an AS-path of 12654 {8210, 20932, 286, 13237, 702, 2497, 9044, 1239, 3320} at 14:55:35 UTC, allowed the discovery of two previously unknown peers of AS 13030 which propagate the announcements for the prefix. A previously performed withdrawal observation had not discovered either of these ASes as direct peers of AS 13030.




(a) (b)

Figure 7: Using targeted announcements to discover the peers of an upstream ISP. (a) Stable state feasibility graph, which shows that AS 13030 is the only upstream of AS 12654 and has 9 visible peers. (b) Feasibility graph obtained by making a single announcement with AS-path 12654 {8210, 20932, 286, 13237, 702, 2497, 9044, 1239, 3320}. The two newly-discovered peers of AS 13030 are in black.

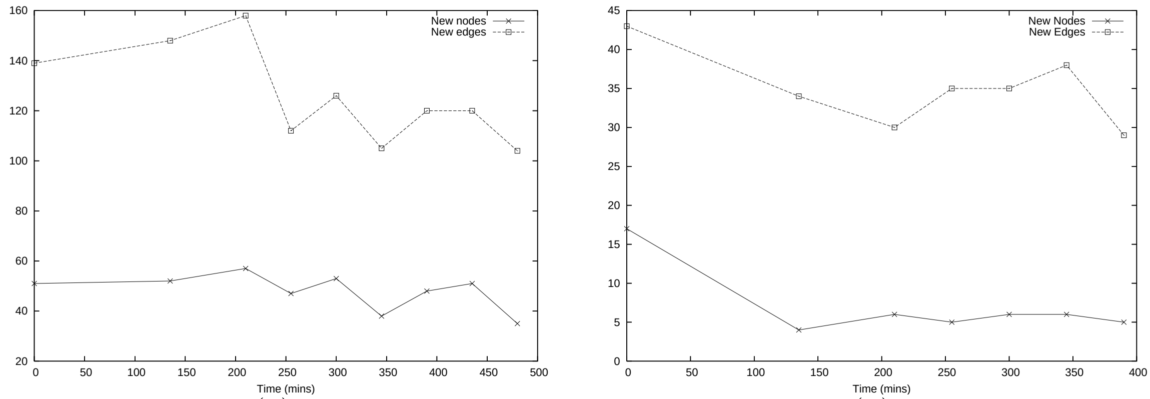
## 6.3 Impact of route-flap dampening

As described in Section 3.3, effective use of our discovery strategies requires that the interval between updates be appropriately determined to avoid the effects of route-flap dampening. To provide a worst-case estimation of the effects on dampening of a single BGP update, we sent out withdrawals at varying time intervals, since a single withdrawal typically generates a very large number of updates due to the path exploration process, thus greatly increasing the likelihood of dampening.

The first withdrawal was sent in a stable routing state, many hours after the last previously observed BGP update for the prefix. The second withdrawal was sent after approximately two hours; the third was sent approximately one hour later, and subsequent withdrawals were sent approximately every half hour. After each withdrawal, we observed the BGP convergence process in a 15-minute interval and measured the number of new ASes and peerings discovered. The prefix was then reannounced in preparation for the next withdrawal.

The IPv6 results in Fig. 8(a) show that the number of new ASes and peerings found substantially decreases if the withdrawal is performed less than one hour after the previous withdrawal. Furthermore, approximately one hour after the first withdrawal, additional BGP activity for the prefix was observed, possibly due to paths that were dampened during the withdrawal being unsuppressed and reused. This suggests that the maximum time that a route is suppressed in the IPv6 Internet is approximately one hour. This agrees with the results in [15].

Our IPv4 results, however, suggests that dampening continues to affect routing for longer than one hour: all subsequent withdrawal observations discovered about 35-40 new peerings and about 5 new nodes, compared with 43 new peerings and 17 new nodes for the first withdrawal. The results suggest that an appropriate interval between updates should be over two hours if the maximum effectiveness of our topology discovery is to be attained.




(a) (b)

Figure 8: Effect of route flap dampening in IPv6 (a) and IPv4 (b). The data points in the graph correspond to withdrawal observations. The x coordinate of each data point represents the instant in which the withdrawal was sent, and the y coordinates represents the number of new ASes and peerings that were discovered with respect to the preceding announcement.

IPv6

| Date | I | W | I only | W only |
|---|---|---|---|---|
| 2005/02/23 09:54 | 312 | 158 (51%) | 175 | 21 (13%) |
| 2005/02/25 10:03 | 334 | 168 (50%) | 189 | 23 (14%) |
| 2005/02/27 15:18 | 302 | 154 (51%) | 174 | 26 (17%) |

IPv4

| Date | I | W | I only | W only |
|---|---|---|---|---|
| 2005/07/05 00:00 | 241 | 61 (25%) | 181 | 1 (2%) |

Table 2: Comparison between the arcs in the graph $W$ generated by withdrawal observation and those in the graph $I$ induced by $W$ in the global AS-graph $C$.

## 6.4 Comparison with the full AS graph

We also compared the results of our per-prefix discovery strategies with more conventional interdomain topology discovery techniques which obtain information by observing the global AS-graph $C$ containing the AS-paths used by all the prefixes in the Internet. At a given time, we simultaneously obtained a feasibility graph $W$ using withdrawal observation and a full AS graph $C$ for all the prefixes announced on the Internet. We then compared $W$ with the graph $I$ induced by the nodes of $W$ in $C$.

The results, in Table 2, show that our per-prefix graphs only have about 50% of the arcs of the induced graphs for IPv6 and 25% for IPv4. This shows that there is a substantial difference between existing topology discovery methods and our active probing discovery methods. The topology captured by the former is much richer; however, the topology discovered by our techniques consists of only those ASes and peerings that may actually be traversed by BGP announcements from $p$ (and thus traffic flows to $p$) and is thus more valuable from an ISP's point of view.

Finally, we note that between 13% and 17% of the arcs in the IPv6 per-prefix graphs were not visible in the graphs induced in the full AS graph. For IPv4, this figure is only 2%. We suspect that this is due to the much greater redundancy of the IPv6 network, in which the widespread use of tunnels makes it easy to create a much denser connectivity mesh.

## 6.5 Path Feasibility Determination

To verify our path feasibility determination techniques, we obtained an initial feasibility graph using withdrawal observation and chose arbitrary paths on the graph starting from the origin AS and ending in a route collector, then applied the nailed-path algorithm to determine which of the paths were feasible. A few examples from our results are in Table 3.

The "Prefix" column shows the prefix we tested and therefore whether the test was performed in the IPv6 or IPv4 network. The "Path" column shows the path we tested. The "UTC Time" column contains the time at which the BGP announcement was sent and the "AS-set" column shows the AS-set announced. The "Observed Path" column shows the path that was observed after BGP propagation and the "Feasible" column shows whether the path was feasible. Note: if no AS-path is observed then we can affirm that a path is not feasible; if another AS-path was observed, it is not possible to determine feasibility (see Section 4.2).

## 6.6 Path Preference Comparison

We tested our path preference comparison technique on various paths ending in route collectors. Table 4 shows some of our results, and Fig. 4 shows a graphical representation of the experiment in the third row.

We note that the second example in our table is interesting because it shows an AS preferring a longer path over a shorter path: between the two paths 1103 2607 1275 15589 5397 and 1103 20965 1299 3320 15589 15589 5397, AS 1103 (SURFnet, the Dutch academic and research network), prefers the latter. This suggests that AS 1103 is explicitly configuring its routers to prefer paths coming from AS 20965 (the Géant European academic and research network), over other paths.

# 7 Conclusions and future work

In this paper we have presented techniques for active probing of the interdomain topology that permit the operators of an ISP to gain a greater understanding of how the ISP's BGP announcements are propagated than by using existing techniques. Namely, we have shown a basic set of probing primitives and discovery strategies based on these primitives, and we have discussed operational impact and ethical issues. Extensive experimental results show the effectiveness of our methods both in the IPv6 and the IPv4 Internet; we believe that our techniques can be very useful for operators and have a negligible impact on bandwidth and router overhead.

This paper opens several new problems:

- The development of new topology discovery algorithms based on our primitives;

| Prefix | UTC Time | Path | AS-set | Observed Path | Feasible |
|---|---|---|---|---|---|
| 2001:a30::/32 | 2005-02-03 18:43:40 | 3257 2497 2500 4691 33 15589 5397 | {4725, 5511, 7660, 18084, 7684, 6939, 10566, 1275, 3320, 5609, 2914, 14277, 6175, 5623, 278, 4697, 3549, 13944, 4555, 15897, 680, 31103, 1299, 5539, 9112, 3246, 8657, 8447, 1257, 4725, 17715, 6435, 145, 12779, 25358, 20965, 1853, 3265, 16713, 109, 6762, 559, 29686, 3344, 8664, 12968, 13110, 8763} | 3257 2497 2500 4691 33 15589 15589 5397 | Yes |
| 2001:a30::/32 | 2005-02-21 17:02:27 | 3333 3265 6175 13977 6939 15589 5397 | {10566, 33, 1275, 3320, 5623, 7580, 6435, 5539, 12477, 4716, 15897, 4697, 5609, 1299, 2607, 31103, 293, 4555, 2042, 8175, 145, 6342, 8447, 3549, 12779, 1257, 3257, 8763, 1752, 2500, 4725, 25358, 3246, 20965, 1853, 559, 1103, 29686, 3245, 513} | 3333 3265 6175 13977 6939 6939 15589 15589 5397 | Yes |
| 2001:a30::/32 | 2005-02-21 16:18:25 | 3333 1103 2607 1275 15589 5397 | {6939, 10566, 33, 3320, 5623, 13944, 7580, 6435, 6175, 5539, 14277, 4716, 15897, 4697, 5609, 2914, 1299, 31103, 293, 4555, 2042, 8175, 145, 6342, 8447, 3549, 12779, 3265, 1257, 3257, 8763, 1752, 2500, 4725, 25358, 3246, 20965, 1853, 559, 29686, 3425, 513, 278, 12859, 8472, 6830, 18084, 2497, 7684, 29377, 1930, 11537, 7660, 5511} | 3333 1103 2607 1275 15589 15589 5397 | Yes |
| 2001:a30::/32 | 2005-02-23 15:27:39 | 6175 4555 13944 6939 15589 5397 | {1275, 3320, 33, 10566, 2607, 109, 5609, 293, 513, 31103, 1299, 2497, 5623, 4725, 17715, 3549, 5539, 14277, 2914, 4697, 4716, 6435, 7580, 3748, 2042, 15897, 2549, 6762, 3425, 9264, 5430, 20965, 1103, 29686, 1752, 1853, 25358, 6830, 559, 3257, 12779, 3265, 2500, 145, 8763, 4691, 3786} | – | No |
| 2001:a30::/32 | 2005-02-23 15:38:30 | 6175 145 7580 10566 15589 5397 | {1275, 3320, 6939, 33, 2607, 109, 5609, 293, 513, 31103, 1299, 2497, 5623, 4725, 17715, 3549, 5539, 14277, 2914, 4697, 4716, 6435, 13944, 3748, 2042, 15897} | – | No |
| 2001:a30::/32 | 2005-04-19 13:56:56 | 559 1299 3320 1275 15589 15589 5397 | {33, 109,145, 278, 293, 513, 559, 1103, 1257, 1752, 1853, 2042, 2497, 2500, 2607, 2914, 3257, 3265, 3292, 3352, 3425, 3549, 3748, 3786, 4691, 4697, 4716, 4725, 5609, 5623, 6175, 6320, 6342, 6435, 6830, 6939, 7033, 8447, 10566, 12779, 13944, 14277, 17715, 17965, 20965, 24136, 24895, 29686, 31103, 32266} | 559 1299 3320 15589 15589 5397 | Unknown |
| 84.205.89.0/24 | 2005-07-05 11:31:44 | 8468 6461 701 702 13030 12654 12654 | {8210, 8220, 3320, 286, 8447, 20932, 9044, 12793, 13237, 2497, 8434, 513, 8289, 8342, 16034, 1239, 209, 3561, 6762, 2914, 1299, 3356} | 8468 6461 13030 12654 12654 | Unknown |
| 84.205.89.0/24 | 2005-07-05 11:36:01 | 8468 6461 13030 12654 12654 | {8210, 8220, 3320, 286, 8447, 20932, 9044, 12793, 13237, 2497, 8434, 513, 8289, 8342, 16034, 1239, 209, 3561, 6762, 2914, 1299, 3356, 701, 702} | 8468 6461 13030 12654 12654 | Yes |
| 84.205.89.0/24 | 2005-07-05 12:31:01 | 8468 9044 13030 12654 12654 | {8210, 8220, 3320, 286, 8447, 20932, 12793, 13237, 2497, 8434, 513, 8289, 8342, 16034, 1239, 209, 3561, 6762, 2914, 1299, 3356, 701, 702, 6461} | 8468 9044 13030 12654 12654 | Yes |
| 84.205.73.0/24 | 2005-07-05 13:29:29 | 2116 1299 1239 13030 12654 12654 | {209, 286, 513, 701, 702, 2497, 2914, 3320, 3356, 3561, 6461, 6762, 8210, 8220, 8289, 8342, 8434, 8447, 8468, 9044, 12793, 13237, 16034, 20932} | – | No |

Table 3: Path feasibility determination results.

| Prefix | UTC Time | Collector | AS-path $\mathcal{P}_1$ | AS-path $\mathcal{P}_2$ | Observed path | Preferred |
|---|---|---|---|---|---|---|
| 2001:a30::/32 | 2005-02-21 16:30:28 | 3333 | 3333 3265 6175 13944 6939 15589 5397 | 3333 1103 3425 293 3320 15589 5397 | 3333 1103 3425 293 3320 15589 5397 | $\mathcal{P}_2$ |
| 2001:a30::/32 | 2005-04-19 12:38:09 | 1103 | 1103 2607 1275 15589 5397 | 1103 20965 1299 3320 15589 5397 | 1103 20965 1299 3320 15589 15589 5397 | $\mathcal{P}_2$ |
| 84.205.89.0/24 | 2005-07-05 12:31:01 | 8468 | 8468 6461 13030 12654 12654 | 8468 9044 13030 12654 12654 | 8468 9044 13030 12654 12654 | $\mathcal{P}_2$ |

Table 4: Path preference comparison results.

- The formal study of which peerings and/or ASes can be discovered by our techniques;
- The combination of AS-set stuffing with network measurements (e.g. RTT probes) to evaluate how network performance would be affected by alternate routing configurations.

We leave these problems to further study.

# Acknowledgements

We would like to thank Maurizio Goretti and Gabriele Barbagallo from CASPUR for providing us with the infrastructure for sending IPv6 BGP announcements from AS5397; this work would not have been possible without their support. We would also like to thank Henk Uijterwaal and Monica Cortes from RIPE NCC for providing us with the opportunity to run tests on Juniper routers and providing direct access to the RIS database during the course of our experiments. Thanks go to Tim Griffin for suggesting the use of BGP communities to mark AS-set stuffing announcements. Finally, we would like to thank Geoff Huston, Ruediger Volk, and Randy Bush for useful discussion.

# References

[1] Antony Antony. RIS observations. In *RIPE 38*, January 2001.

[2] R. Chandra, P. Traina, and T. Li. RFC 1997: BGP communities attribute, August 1996.

[3] Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott J. Shenker, and Walter Willinger. Towards capturing representative as-level internet topologies. In *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 280–281, New York, NY, USA, 2002. ACM Press.

[4] Xenofontas Dimitropoulos, Dmitri Krioukov, and George Riley. Revisiting Internet AS-level topology discovery. In *Proc. PAM 2005*, pages 177–188, April 2005.

[5] K. Egevang and P. Francis. RFC 1631: The IP network address translator (NAT), May 1994. Status: INFORMATIONAL.

[6] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, Dec 2001.

[7] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of IEEE INFOCOM 2000*, pages 1371–1380, Tel Aviv, Israel, March 2000. IEEE.

[8] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10(2):232–243, 2002.

[9] Bradley Huffaker, Daniel Plummer, David Moore, and k claffy. Topology discovery by active probing. In *Symposium on Applications and the Internet (SAINT)*, http://www.caida.org/outreach/papers/2002/SkitterOverview/, January 2002.

[10] Geoff Huston. Interconnection, peering and settlements – part 1. *Internet Protocol Journal*, 2(1):2–16, 1999.

[11] Geoff Huston. Interconnection, peering and settlements – part 2. *Internet Protocol Journal*, 2(2):2–23, 1999.

[12] Geoff Huston. BGP routing table analysis reports. http://bgp.potaroo.net/, April 2005.

[13] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *ACM SIGCOMM 2000*, pages 175–187, sep 2001.

[14] Z. Mao, R. Bush, T. G. Griffin, and M. Roughan. Bgp beacons. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM Press, 2003.

[15] Z. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates internet routing convergence, 2002.

[16] Z. Mao, J. Rexford, J. Wang, and R. Katz. Towards an accurate AS-level traceroute tool. In *Proc. ACM SIGCOMM 2003*, August 2003.

[17] Zhuoqing Morley Mao, David Johnson, Jennifer Rexford, Jia Wang, and Randy Katz. Scalable and accurate identification of AS-level forwarding paths. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 2004. IEEE.

[18] RouteViews project, university of Oregon. `http://www.routeviews.org/`.

[19] Y. Rekhter. A border gateway protocol 4 (BGP-4). IETF, RFC 1771.

[20] Routing Information Service of the RIPE (RIS). `http://www.ripe.net/ripencc/pub-services/np/ris/`.

[21] Roma Tre Computer Networks research group. Active bgp probing web site. On-line `http://www.dia.uniroma3.it/~compunet/bgp-probing/`.

[22] Roma Tre Computer Networks research group. Bgplay. http://bgplay.routeviews.org/bgplay/, http://www.ris.ripe.net/bgplay/.

[23] W. Simpson. IP in IP tunneling. RFC 1853, October 1995.

[24] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with rocketfuel. In *Proc. ACM/SIGCOMM '02*, pages 133–145, August 2002.

[25] W. Stevens. RFC 2001: TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, January 1997. Status: PROPOSED STANDARD.

[26] John W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, Reading, MA, 1999.

[27] Jianhong Xia. In *RIPE Routing Working Group mailing list archive*, June 2002.

[28] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the internet as-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1):53–61, 2005.