New algorithmic tools for inter-domain topology discovery based on active BGP probing.

Integration of AS relationship classification in the prototypical platform realized as D1.1.2
Start date of the project: January 2004
Duration: 48 months
Project Coordinator: Prof. Dr. math. Friedhelm Meyer auf der Heide
Heinz Nixdorf Institute, University of Paderborn, Germany

Due date of deliverable: December 2005
Actual submission date: January 2006
Dissemination level: PU – public

Work Package 1.1: Monitoring, Visualizing, and Analyzing Large Dynamically Evolving Information Systems

Participants: Università Degli Studi di Roma "La Sapienza" (UDRLS), Italy
ETH Zürich, Switzerland

Authors of deliverable: Alexander Hall (alex.hall@gmail.com)
Maurizio Pizzonia (pizzonia@dia.uniroma3.it)
Contents

1 Introduction 2

2 Topographic Visualization of the Internet 2
   2.1 State of the Art .................................................. 3
   2.2 The Topographic Visualization Metaphor ............................. 3
   2.3 The Visualization Algorithm ...................................... 5

3 Active BGP Probing 5
   3.1 State of the Art .................................................. 7
   3.2 BGP Active Discovery Algorithms ................................. 9
   3.3 Experimental Results ............................................. 9

4 Topology Discovery with Distance Queries 10
   4.1 State of the Art .................................................. 10
   4.2 Results ............................................................. 11
1 Introduction

For an Internet Service provider (ISP), the knowledge of which paths are traversed by its traffic flows is essential to predict the impact of network faults, to perform effective traffic engineering, to develop peering strategies with other ISPs, and to assess the quality of connectivity provided by the upstream providers from which the ISP buys connectivity to the rest of the Internet.

Since Internet is not a static, centrally planned, network, there is no central authority that can help ISPs providing a map of it. In this context, research in network discovery and network visualization are important for supporting the strategic decision process of ISPs.

The work described in this report contributes to the algorithmic tools for network discovery and visualization in the following ways.

- To visually represent routing paths traversed by traffic flows we propose a topographic metaphor which highlights the levels of the hierarchy they traverse and hence the “expensiveness” of the path. We propose an algorithm to compute such Internet topographic maps and show preliminary results. This work is extensively described in [10] (attached to this deliverable). The algorithm has been implemented in the BGPlay routing visualization system which is a web based monitoring tool (developed within the DELIS project as part of Deliverable 1.1.2) and widely used by ISPs [8, 32].

- 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. This work is extensively described in [9] (attached to this deliverable).

- We study the problem of finding the minimum set of queries to discover all edges and non-edges of the network graph. We propose a distance query model and, within this context, we study the on-line discovery problem using competitive analysis and give a randomized on-line algorithm with competitive ratio $O(\sqrt{n \log n})$ for graphs on $n$ nodes. We also show lower bounds on competitive ratios of deterministic on-line algorithms and randomized on-line algorithms, respectively. In the off-line network verification problem, the graph is known in the beginning and the problem asks for a minimum number of queries to verify all edges and non-edges. We show that the problem is $NP$-hard and present an $O(\log n)$-approximation algorithm. This work is extensively described in [18] (attached to this deliverable). The work presented in [18] is also partially reported as Deliverable 1.1.3 of the DELIS project.

This report is organized as follows. In Section 2 we summarize the relevant results of the visualization of routing paths with the topographic metaphor that we implemented in the BGPlay system. In Section 3 we describe the active BGP probing discovery technique we devised and summarize the main experimental results. In Section 3 we provides the results about topology discovery using distance queries. State of the art is provided in each of these sections.

2 Topographic Visualization of the Internet

In the Internet thousands of Internet Service Providers (ISPs) exchange traffic according to their commercial agreements. ISPs are usually thought as organized in a hierarchy with the most important
at the top. The peculiar business model of the Internet makes paths that traverse higher levels of the hierarchy more expensive than paths that traverse only lower levels.

To visually represent routing paths traversed by traffic flows we propose a topographic metaphor which highlights the levels of the hierarchy they traverse. We propose an algorithm to compute such Internet topographic maps and show preliminary results. Our implementation is based on the BGPlay routing visualization system which is a web based monitoring tool widely used by ISPs.

2.1 State of the Art

If we are interested in the relationships between Internet Service Providers (ISP) we can look at Internet at the Autonomous Systems (AS) level, where an AS is a collection of networks under the administrative authority of the same ISP. At such a level, Internet is currently partitioned into more than 20,000 ASes.

Many efforts have been done in the literature for visualizing the Internet at the AS level. A survey about the visualization of the Internet can be found in [16]. Some systems aim at visualizing the entire Internet structure (see e.g.,[24, 3, 11, 1, 2]), while others visualize a little portion of it with different specific purposes. For example in [6] is described a system that visualizes the relationships between ASes that are known to an Internet Registry. Concerning “live” routing data, the BGPlay system [8] displays the evolution of the routing paths toward a specific AS over time. BGPlay is a quite popular tool for the ISPs and has been adopted by international organizations that publish real time collected data about inter-AS routing [30, 29]. A project that has a similar intent is shown in [28].

Recent studies [22] and [13] (the latter developed within the DELIS project as part of Deliverable 1.1.2) have shown that the ISPs (and then the ASes) of the Internet are arranged into a customer-provider hierarchy (in the following Internet Hierarchy). In such hierarchy customers buy from their providers a transit service for their traffic to and from the Internet. In [36] each ISP is assigned to a level, according to its rank in the Internet Hierarchy. At this point, the problem of visually represent the Internet Hierarchy naturally arises.

An amazing number of methods have been devised to visually represent hierarchies in a variety of application domains (see, for example, [31, 33, 4]) many of which are targeted to show very large hierarchies.

The visualization of the structure of the Internet and of the Internet Hierarchy can be interpreted as a special case of a clustered graph drawing problem (first introduced in [19]), where the hierarchy describes a simple structure of clusters. Several authors dealt with the problem of representing a clustered graph using a spring embedder approach. Eades and Huang [17] proposed a system for the visualization of huge graphs, by first performing a clustering and then visualizing a portion of the graph by applying a force-directed approach. Walshaw [37] introduced an heuristic method for drawing large graphs which uses a multilevel technique combined with a force directed placement algorithm. Frishman and Tal [20] proposed an algorithm for dynamic drawings of clustered graphs.

2.2 The Topographic Visualization Metaphor

Fig. 1 shows a screenshot of the BGPlay system without the topographic map enhancement. Each number represents an AS, and the AS originating the prefix is placed in the center of the graph and highlighted by a red circle (in this case AS137). Each solid or dashed line represents a segment of an AS-path seen by RIS or Oregon Route Views collectors. An AS-path starts in the originating AS and stops in the AS which provides BGP routing data to RIS or Oregon Route Views. The picture represents paths to reach AS137 from several other ASes at a certain instant of time. For example, the path AS16150, AS6939, AS6762, AS137 is shown to be used for traffic incoming AS137 from AS16150. The AS-paths that did not change during the time interval specified by the user are
Figure 1: A snapshot of the BGPlay system.

Figure 2: A screenshot of the BGPlay system enhanced by the visualization approach described in this paper.
merged into trees rooted at the origin AS and drawn dashed. Each tree is drawn in different color so that it can be unambiguously identified. AS-paths that did change during the query interval are drawn solid and are not merged. No information is available in this visualization about the relevance of ASes and their economic relationships.

Fig. 2 shows the same map as visualized by BGPlay enhanced with the topographic visualization approach presented in [10]. The contour lines are used to confine ASes that are at the same level of the hierarchy. For example, the ASes inside the central brown area are top level ASes, that is they are part of the Internet backbone. As the “mountain” decreases in height the ASes decrease their rank in the hierarchy. The map shows quite well that some of the traffic flows have to climb the entire hierarchy to reach AS137, while other flows take “shortcuts”. Consider the path of the example above: AS16150, AS6939, AS6762, AS137. From the picture, it is easy to see that it does not pass through the Internet backbone, since AS6939 and AS6762 exchange traffic at a lower level. Such kind of paths are usually more efficient and less expensive than paths that pass through the Internet backbone.

Concerning the representation of the AS-paths and of the ways they change over time, we adopt the same drawing standard adopted by BGPlay. Further details are provided in [10].

2.3 The Visualization Algorithm

The layout algorithm is based on a variation of the well-known force directed approach that is usually called spring embedder (see for example [21, 12]). In a conventional spring embedder, the layout of a graph is obtained by simulating a physical system in which its vertices are electric charges that can move in the plane and are subject to physical forces. Each charge is repulsed by all other charges. Each edge is a spring which has a length “at-rest” \( l \). Each edge attracts its extremes if their distance is greater than \( l \) and repulses them if their distance is less than \( l \). The placement of the charges in the plane at equilibrium describes the final layout of the graph.

Our layout algorithm acts in two phases. In the first phase, vertices are properly placed within concentric circles called fences representing the contour lines that separates the layers. This is performed with an incremental approach starting from the center of the drawing (see Figures 3a-d). In the second phase the fences can change their shapes to adapt to the graph layout. They are represented by cycles made of vertices placed very close to each other so that other vertices of the graph cannot traverse the fence. To avoid that fences intersect each other and that ASes broke the confinement of the fences, the density of the vertices of the fences is kept within a safe range by dynamically adding or removing vertices as the fences change their perimeters. The result of this phase is shown in Figure 3e and the corresponding picture that is presented to the user by the enhanced version of the BGPlay system is shown in Figure 2).

3 Active BGP Probing

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 distance three the knowledge of how the network treats \( Z \)’s prefixes becomes negligible.

Consider, as an example, the drawing in Fig. 4. Fig. 4(a) shows what an operator of AS 5397 may discover about the routing of its IPv6 prefix 2001:a300::/32 on Dec 30 2004 at 02:44:00 UTC by querying the RIPE NCC RIS service [30]. An arrow from an AS \( A \) to an AS \( B \) in Fig. 4(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
Figure 3: An example of execution of the layout algorithm for topographic visualization of the Internet. The final result presented to the user is shown in Figure 2.
AS 33, but cannot see, for example, that AS 15589 is propagating the announcement to AS 1275 as well.

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. 4(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. Fig. 5 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 the knowledge of ASes and peerings traversed by the prefix is immediately apparent.

3.1 State of the Art

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 analyzing BGP routing tables [7] and announcements [32], and various techniques have been proposed to deduce the interdomain structure using probe packets [24, 35, 26, 23, 27]. Recently, BGP beacons [25] 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 [38, 15], 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.
Figure 5: What an operator can see using the techniques described in this paper, in the same situation as in Fig. 4.
3.2 BGP Active Discovery Algorithms

In this section we give a brief sketch of our algorithms, further details can be found in [9] (attached to this report).

The main primitive used by our approach is \textit{AS-set stuffing}. It consists in announcing prefix $p$ with an AS-path of $Z\{A_1A_2\ldots A_n\}$, where $\{A_1A_2\ldots A_n\}$ is an AS-set. The presence of the AS-set prevents BGP announcements for $p$ from traversing an arbitrarily chosen set of ASes $A_1, \ldots, A_n$. We name these ASes \textit{prohibited} ASes. 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.

The second primitives is \textit{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.

An AS $Z$ may obtain an AS-graph at a given time using a query to the route collectors [30, 29], but the extent of such a graph is limited, as can be seen in Fig. 4. Our strategies for prefix propagation discovery permit $Z$ to obtain a much larger feasibility graph. AS $Z$ may use withdrawal observation to obtain a large AS-graph and then use of AS-set stuffing to refine the search.

We begin with the directed AS-graph seen by the route collectors at a certain instant and proceed level by level, starting from level one. The level of an AS is its topological distance from $Z$ in the directed AS-graph. 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. 4(b) shows the new nodes and arcs discovered starting from the situation in Fig. 4(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. 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 similar approach can be used to know whether a certain AS-path $P$ is feasible for a prefix $p$. For this purpose, we consider the AS-path $Q$ seen by a route collector to reach $p$ and use AS-set stuffing to prohibit the ASes in $Q$ which differs from those in $P$.

AS-set stuffing can be also used to determine which of two AS-paths $P_1$ and $P_2$ ending in the same AS $A$ is preferred by $A$. We prohibit all the ASes in all levels up to the level of $A$, except the ASes in $P_1 \cup P_2$. Usually, this is enough for $A$ to see either $P_1$ or $P_2$, that is the AS-path preferred by $A$. Further details can be found in [9]. An example of this application is shown in Figure 6.

3.3 Experimental Results

To evaluate the effectiveness of our topology discovery strategies, we performed a large number of experiments in both IPv4 and IPv6 network. We compared the results of observation of stable routing state, of withdrawal observation, and of our level-by-level exploration. The level-by-level approach performed better than the other two. We studied the effect of route flap dampening on our techniques. We compared the results of our per-prefix level-by-level discovery with more conventional interdomain topology discovery techniques which obtain information by observing the global AS-graph containing the AS-paths used by all the prefixes in the Internet. We experimented the techniques proposed for inferring the feasibility of a certain AS-path and the preference between two AS-paths by a certain
4 Topology Discovery with Distance Queries

The network discovery (verification) problem asks for a minimum subset $Q \subseteq V$ of queries in an undirected graph $G = (V, E)$ such that these queries discover (verify) all edges and non-edges of the graph. This is motivated by the common approach of combining local measurements in order to obtain maps of the Internet or other dynamically growing networks. In the distance query model, a query at node $q$ returns the distances from $q$ to all other nodes in the graph. We describe how the existence of an individual edge or non-edge in $G$ can be deduced by potentially combining the results of several queries. This leads to a characterization of when a set of queries $Q$ “discovers” the graph $G$. In the on-line network discovery problem ($\text{Dist\textendash ALL\textendash DISCOVERY}$), the graph is initially unknown, and the algorithm has to select queries one by one based only on the results of the previous ones.

In [18] we study the problem using competitive analysis and give a randomized on-line algorithm with competitive ratio $O(\sqrt{n \log n})$ for graphs on $n$ nodes. We also show lower bounds $\Omega(\sqrt{n})$ and $\Omega(\log n)$ on competitive ratios of deterministic on-line algorithms and randomized on-line algorithms, respectively. In the off-line network verification problem ($\text{Dist\textendash ALL\textendash VERIFICATION}$), the graph is known in the beginning and the problem asks for a minimum number of queries to verify all edges and non-edges. We show that the problem is $\mathcal{NP}$-hard and present an $O(\log n)$-approximation algorithm.

4.1 State of the Art

There are several ongoing large scale efforts to collect data representing local views of the Internet, here we will only mention two. The most prominent one is probably the RouteViews project [29] by the University of Oregon. It collects data from a large number of so called border gateway protocol routers. Essentially for each router—which can be seen as a node in the Internet graph—the list of paths it knows (to all other nodes in the network) is retrieved. More recently, and due to good publicity very successfully, the DIMES project [14] has started collecting data with the help of a volunteer community, similar in spirit to SETI@Home [34]. Users can download a client which collects paths in the Internet by executing successive traceroute commands. A central server can direct each client individually by specifying which routes to investigate.
Data obtained by these or similar projects has been used in heuristics to obtain maps of the Internet, basically by simply overlaying possible paths found by the respective project, see e.g., [23, 29, 14, 1]. Another line of research aims at inferring from such local views the types of economic relationships between nodes in the Internet graph, cf. [22, 36] and [13] which is part of deliverable D1.1.2.

Beerliova et al. [5], reported on in deliverable D1.1.1, propose the general problem of network discovery (verification) and study it for the “layered graph” query model: a query \( q \in V \) returns all edges and non-edges between nodes of different distances from \( q \). They present an \( o(\log n) \) inapproximability result for the off-line version and give a randomized on-line algorithm with competitive ratio \( O(\sqrt{n \log n}) \). The on-line algorithm presented in [18] is based on a similar approach, but requires new ideas.

4.2 Results

In [18] we give a characterization of the queries that discover an individual non-edge and the sets of queries that together discover an individual edge. These characterizations are important building blocks of our work.

For general graphs, the Dist–All–Verification problem—off-line: verify a given graph with a minimum number of queries—turns out to be \( \mathcal{NP} \)-hard. This result is shown in Section 4.2 of [18]. An \( O(\log n) \)-approximation algorithm for the same problem is presented in Section 4.3 of [18].

For Dist–All–Discovery—on-line: discover a graph, query by query—we show in Section 5.1 of [18] that no deterministic on-line algorithm can be better than \( O(\sqrt{n}) \)-competitive and no randomized on-line algorithm can be better than \( O(\log n) \)-competitive. Finally, we present our main result: a randomized on-line algorithm with competitive ratio \( O(\sqrt{n \log n}) \). This result is presented in Section 5.2 of [18].

References


[34] SETI. Seti@home website, 2005. http://setiathome.ssl.berkeley.edu/.


Topographic Visualization of The Internet

P. F. Cortese, G. Di Battista, A. Moneta, M. Patrignani, and M. Pizzonia

RT-DIA-104-2005 Dicembre 2005

Dipartimento di Informatica e Automazione,
Università di Roma Tre,
Rome, Italy.
{cortese,gdb,patrigna,pizzonia}@dia.uniroma3.it
antonello.monetsen@gmail.com