Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 9, no. 1, pp. 117–148 (2005)

Visualizing Interdomain Routing with BGPlay

Lorenzo Colitti *Giuseppe Di Battista* *Federico Mariani*
Maurizio Patrignani *Maurizio Pizzonia*

Dipartimento di Informatica ed Automazione
Università di Roma Tre
http://www.dia.uniroma3.it/
{colitti,gdb,mariani,patrigna,pizzonia}@dia.uniroma3.it

Abstract

In this paper we describe the architecture and the visual interface of BGPlay, an on-line service for the visualization of the behavior and of the instabilities of Internet routing at the autonomous system level.

A graph showing only the connections among autonomous systems is not enough to convey all the information needed to fully understand routing and its changes. BGPlay uses specifically tailored techniques and algorithms to display the state of routing at specific points in time and to animate its changes. The system obtains routing data from well known on-line archives of routing information which are constantly kept up-to-date.

Article Type	Communicated by	Submitted	Revised
regular paper	G. Liotta	May 2004	July 2005

Preliminary extended abstracts describing different parts of this work were published by Di Battista, Mariani, Patrignani, and Pizzonia in the Proceedings of the International Workshop on Inter-domain Performance and Simulation (IPS 2003) and in the Proceedings of the 11th International Symposium on Graph Drawing (GD 2003).

Work partially supported by European Commission - Fet Open project DELIS IST-001907 "Dynamically Evolving Large Scale Information Systems", by MIUR 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.

1 Introduction

The Internet is administratively partitioned into networks, called *Autonomous Systems* (*AS*), where each AS is under a single administrative authority. Usually, an Internet Service Provider (ISP) controls one or more ASes.

Roughly speaking, at the AS level Internet routing is described by a collection of sequences of ASes, called *AS-paths*, each one of them associated with a contiguous portion of Internet address space, called *prefix*. An AS-path $AS1, AS2 \ldots, ASn$ associated with prefix P_1 says that the packets directed to addresses in P_1 and originated by some device in $AS1$ should traverse $AS2$, $AS3$, etc. in the specified order, until they reach ASn, which contains the devices with addresses in P_1. This high-level routing information, which is essentially a collection of paths changing over time, is what is found in the routing tables of ASes' border routers. Also, historic and up-to-date routing information, collected from several vantage points of the Internet, is managed by large and well known repositories such as the Oregon Route Views (ORV) [18] project and the RIPE NCC's Routing Information Service (RIS) [20]. The collection of all the AS-paths in use in the Internet at a given moment can be "merged" into a graph, called the *AS graph*. The AS graph is a huge graph: the number of ASes participating in the interdomain routing process is above 15,000, and each of them uses one or more distinct AS-paths to route packets toward each of the (currently over 130,000) prefixes in the Internet.

Exploring, monitoring, and visualizing the interdomain routing is an important, albeit elusive, goal. There is no central authority that determines routing in the Internet, and it is difficult to anticipate the combined effects of many independent administrations continuously making routing changes to reduce costs and optimize local efficiency. Also, it is not known how to enforce stability in the presence of uncoordinated policies or faults of links and routers [14, 8, 6, 13]. Several works address the problem of exploring and visualizing the Internet at the level of the routers involved in the forwarding process (see, for example, [7, 10, 21]), and some visualization techniques for representing the Internet at the level of Autonomous Systems have been applied to the specific problem of visualizing the information stored in the routing registries [3, 9]. However, to our knowledge, no tool is available to represent the evolution of the AS graph.

Obviously, a detailed visualization of the whole AS graph is likely to be incomprehensible. Fortunately, since in the Internet every prefix is routed independently, it is possible to reconstruct meaningful portions of the whole graph by combining the results of single-prefix explorations. Furthermore, the information sought by potential users of such a visualization tool often concerns the behavior of only a single prefix. For example, the Network Operating Center (NOC) of some ISP may be interested in the AS-paths used to reach a prefix under the authority of the NOC itself, or a user may be interested in the AS-path used to reach a prefix to which connectivity is poor. Hence, examining the routing of a single prefix at a time is a natural strategy which meets practical needs as well as substantially simplifying the problem.

By querying RIS or ORV it is possible to obtain all the available routing information about a prefix in a given time window, but without a graphic representation it is not easy to understand how the AS-paths intersect and the big picture that is obtained by merging them together. At the same time, any visualization tool must cope with the peculiarities of routing information: any AS-path must be unambiguously recognizable from the whole picture and its changes over time must be clearly represented.

BGPlay displays the portion of the AS graph that describes how traffic flows to a specific prefix from a selected set of ASes which are considered, in some sense, "representative" of the entire Internet. BGPlay also provides a visual representation of the evolution of this portion of the graph during a specified time interval. This can be important for several reasons: to detect faults in the links traversed by the traffic flows, to check the consistency of router configurations, or even to monitor the behavior of ASes participating in the routing process in order to verify whether they fulfill commercial agreements. It is especially useful to verify reports of transient connectivity problems that occurred in the past but are no longer present.

In this paper we describe the architecture and the visual interface of BGPlay, which can be found at `http://www.dia.uniroma3.it/∼compunet/bgplay/`. Two instances of BGPlay, providing better performance and access to more data, have also been deployed by the RIS and by the ORV.

BGPlay has a client-server architecture. The user performs a query on a Web browser (the client side), specifying an IP prefix and a time interval. The server identifies the AS that contains the given prefix and extracts information that describes the routing evolution in the given interval from the RIS and from a local mirror of ORV. All visualization problems are addressed on the client side by means of a Java applet. The user interface provided by the BGPlay client can visualize the state of the routing graph for a particular prefix at any instant of the query interval, showing the ASes involved and which AS-paths are active for that prefix at a given point in time.

The paper is organized as follows: Section 2 presents basic interdomain routing concepts. Section 3 describes the data sources which BGPlay uses, Section 4 describes the architecture of the BGPlay service itself, and Section 5 shows how the information obtained from the data sources is used to produce a consistent view of the evolution of a particular prefix's routing in a particular time interval. In Section 6 we introduce the concept of *routing graph*, discuss the requirements for its visualization and the visualization approaches we considered, and illustrate the approach adopted by BGPlay; in Section 7 we detail the algorithms themselves. Finally, Section 8 describes BGPlay's user interface and provides examples of how it can be used, both for research purposes and from an operational perspective.

2 Networking Background

In the Internet each host is identified by an IP address (32 bits, usually written in the dotted notation, e.g. 193.204.161.48). An IP prefix identifies a set of (contiguous) IP addresses having the same leftmost n bits, with $0 \leq n \leq 32$, and is usually written by attaching a $/n$ at the end of the prefix (e.g. 193.204.0.0/15 indicates a prefix 15 bits long) [23]. Similarly to telephone call routing, interdomain routing in the Internet is based on the destination prefix. Since a prefix identifies a set of addresses, it implicitly identifies a set of hosts having such addresses. In the following the term *prefix* is used to denote both a set of addresses and a set of hosts; context will serve to disambiguate between the two.

An *Autonomous System* (*AS*) is a portion of the Internet under a single administrative authority. In the Internet each AS is identified by an integer number. ASes cooperate in order to ensure good connectivity service to their customers but are competitors from a commercial point of view [11, 12].

Traffic starting from an AS and directed to a specific prefix traverses an ordered set of ASes (*AS-path*). The configuration of such paths on the routing devices is too complicated to be manually performed. Hence, ASes exchange routing information with other ASes by means of a routing protocol called Border Gateway Protocol (*BGP*) [19, 22]. This protocol is based on a distributed architecture where *border routers* that belong to distinct ASes exchange the information they know about reachability of prefixes. Two border routers that directly exchange information are said to have a a *peering session* between them, and the ASes they belong to are said to be *adjacent*.

Each router stores information about routing in its *routing information base* (*RIB*). The RIB is a table where each row is a pair $\langle \text{prefix}, \text{AS-path} \rangle$ meaning that a certain prefix is reachable through the associated AS-path. Such pairs are called *routes*. The main purpose of BGP is to allow the routers to exchange the routes they know. Since RIBs may be huge, the BGP process running on a router sends to its peers the full RIB only when a peering session is set up. During regular operation only changes to the RIB, termed *updates*, are sent.

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”.

A router which receives an update may or may not modify its routing table, depending on whether or not it knows routes which BGP considers “better” and depending on the routing policy of the AS itself. If the router modifies its routing table, it propagates the update to its peers.

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). These routes are propagated by means of route announcements to adjacent ASes, which in turn propagate it to adjacent ASes. Every time a router propagates an announcement, it prepends its AS identifier to the AS-path; thus, the AS-path of an update is the list of ASes that the update has passed through.

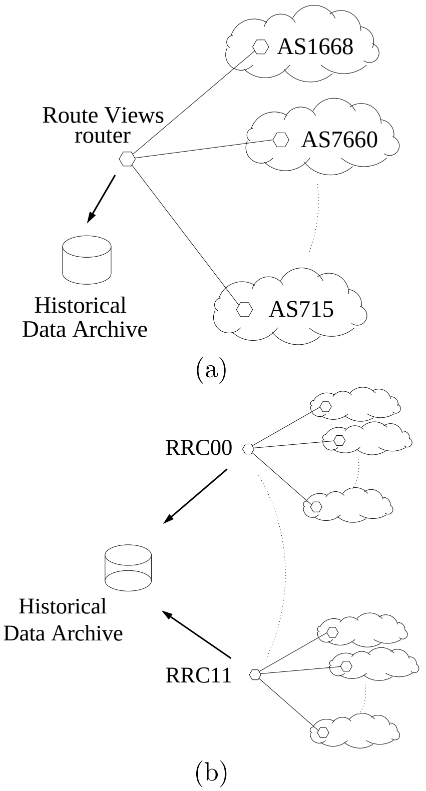


Figure 1: The architecture of ORV (a) and of RIS (b).

3 Data Sources

The Oregon Route Views (ORV) project (see Fig. 1(a)) provides a service for Internet operators to obtain real-time information about the global routing system from the perspectives of several different locations around the Internet. Currently, the Route Views BGP router has more than 60 BGP peering sessions with routers of Internet service providers.

ORV collects BGP updates from its peers without using the routes learned to forward traffic or reannouncing them to other peers. The updates allow us to determine the routes used by each peer at any given moment and follow their evolution over time. ORV data archives contain both snapshots of the global routing information base of ORV at different instants of time and sequences of BGP updates. The data are made available in the MRT [17] format.

The Routing Information Service (RIS) provided by the RIPE NCC (see Fig. 1(b)) collects historical information about Internet routing by using Remote Route Collectors (RRCs) at different locations around the world. This information is integrated into a comprehensive view. A RIS RRC is a BGP router that collects BGP routing information in the same way as the ORV router. Each RRC is identified by a string in the format RRCxx, where xx is a two digit number.

The RIS offers several query facilities which can help network operators troubleshoot routing problems and network researchers study Internet routing and its changes. For example, the BGP Routing Hot Spot Utility generates lists of prefixes, originating from a specified AS, for which high BGP announcement activity has been observed by some RRC. Also, the ASInuse facility determines when an AS number last appeared in the global routing table collected by the RIS.

In this paper, we are interested in a simpler RIS query, that can be performed using the Search by Prefix utility. It allows the user to search the RIS database by specifying a prefix, a time interval, and a set of RRCs, and outputs a list of BGP updates recorded by the selected RRCs in the specified time interval. It also outputs the status of the Local Routing Information Base (Loc-Rib) for the selected prefix in an instant of time that is “related” to the specified time interval.

4 System Architecture

The architecture of BGPlay is based on the following design choices.

- The user interacts with the system by means of a Web browser. A query identifies a time interval and a prefix to be monitored. The data sources to be used can be selected from a set of possible alternatives.

- The results of the query are graphically visualized on the client, which displays the routing graph. The user can display the state of the graph at any time within the query interval and view routing changes, which are displayed using animations. Specific techniques are used to improve the readability and the appearance of the graph: the requirements specific to this problem are described in Section 6 and the graph drawing techniques used are described in Section 7.

- The service is based on a client-server architecture, where the server computes the result of queries and the client is an applet running on the user’s browser. We decided to use an applet instead of producing standard JPEG or GIF images on the server both because of the graphical complexity of the animation and to allow fast interactive access by the user.

- The BGP data used to answer the user queries are obtained from data sources in various ways, including using the RIS web interface, accessing the RIS database directly (for the instance of BGPlay deployed at the RIPE NCC), and mirroring ORV data into a local database. All the data sources have the same semantics, but the method of accessing them differs. The BGPlay server has a modular architecture to facilitate the configuration and addition of new data sources.

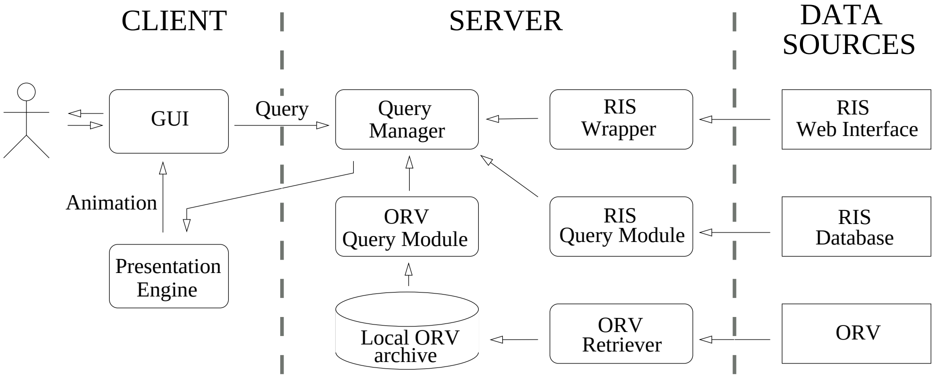


Figure 2: The system architecture.

Fig. 2 illustrates the main components of the architecture:

Graphical User Interface. Provides the user with tools for interacting with the visual representation of routing information. For example: a *time panel* shows the time location of the most important events in the selected time interval, *animation buttons* allow the user to step over events, and an *event display* shows detailed information about the currently visualized event. It also provides a query window where the user can select the prefix to query, the time interval, and the data sources desired.

Presentation Engine. Computes the layout of the routing graph and smoothly animates its layout in response to routing events. It makes use of Graph Drawing methodologies and techniques.

Query Manager. Receives a query from the User Interface, accesses the data sources, computes the result of the query, and delivers the result to the Presentation Engine.

RIS Wrapper. A wrapper that queries the RIS Search by Prefix web interface and retrieves the corresponding results in terms of BGP routing tables and updates.

RIS Query Module. Obtains BGP routing tables and updates using SQL queries to the RIS database. Used by the RIPE NCC instance of BGPlay.

ORV Query Module. Obtains BGP routing tables and updates using SQL queries to the local ORV Archive.

ORV Retriever. Periodically retrieves routing tables and updates available at the ORV raw data archives in the MRT format. Adds the new data to the Local ORV Archive and removes the oldest data from it.

Local ORV Archive. A relational database (currently based on MySQL technology) which stores the ORV data.

An analysis of the workload and of the generated traffic led us to a decomposition of the system in which the Presentation Engine is located on the client side, together with the Graphical User Interface.

As regards time representation, both the RIS and ORV use UTC (Coordinated Universal Time). BGPlay conforms to that convention.

5 Query Processing

In order to answer a client query, the Query Manager must provide the RIBs at the start of the query interval and all the updates that occurred during the query interval. As described in Section 4, three types of retrieval are currently involved: retrieval from the RIS web interface through the RIS Wrapper, retrieval from the RIS database using the RIS Query Module, and retrieval from the Local ORV archive using the ORV Query Module.

Since the data sources used do not permit the query of RIBs at arbitrary points in time, but only provide periodic snapshots of the RIBs themselves, the Query Manager must calculate the RIBs at the start of the query interval by starting from a previous RIB provided by the data source and applying all the BGP updates between that time and the start of the query interval. Consider Fig. 3, where the starting and ending instants of the time interval are called t_1 and t_2, respectively. The purpose of the Query Manager is to compute:

- the status of the ORV RIB in t_1;
- the status of the RIS RIB in t_1; and
- the sequence of updates collected by ORV and RIS between t_1 and t_2.

In general it is unlikely that both RIBs at time t_1 are known. The Query Manager solves this problem by obtaining the RIB at some time $t_0 \leq t_1$ and applying to it all the updates between t_0 and t_1 to derive the RIB at time t_1. More specifically, given the RIB at t_0, it selects the entries for the queried prefix p, applies all the updates to prefix p to the resulting RIB subset, and obtains the subset of the RIB at time t_1 corresponding to prefix p.

The exact means used to obtain the RIB at time t_0 depend on the data source. ORV takes a snapshot of its RIB approximately every two hours. The RIB at time t_0 can then simply be obtained from the last snapshot before time t_1. The situation for RIS is slightly more complex. Although the RIS takes a snapshot of the RIB three times a day (at 00:00, 08:00 and 16:00 UTC), data from the three snapshots is stored in the same database table without distinguishing between RIB entries collected at different times. Thus, a RIB query on a given day returns precisely the data from a snapshot only if the query is performed between 00:00 and 08:00 and if the query requests the RIB

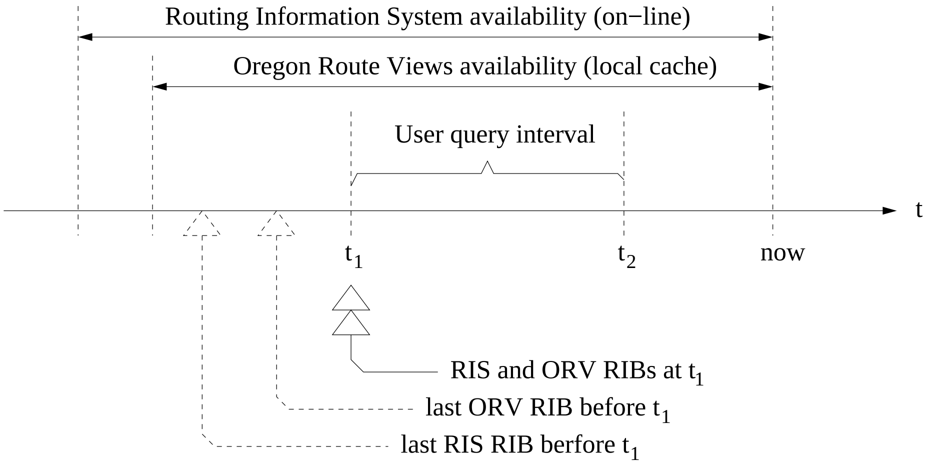

Routing Information System availability (on−line)
Oregon Route Views availability (local cache)
User query interval
t
t_1
t_2
now
RIS and ORV RIBs at t_1
last ORV RIB before t_1
last RIS RIB berfore t_1

Figure 3: Available and computed RIBs.

on the same day. In all other cases, duplicate entries are present. However, the RIS database does store the time when a RIB entry was last modified. Thus, the Query Manager chooses t_0 as 00:00 on the same day as t_1 and obtains the RIB at time t_0 by performing a query for all the RIB records whose last update occurred before t_0.

The Query Manager then combines all the computed RIBs (one for each data source) into a single RIB computed at time t_1 and sends the result to the presentation engine together with a list of all the updates in the time interval.

6 Visualization Requirements and Choices

The purpose of BGPlay is to provide a graphical representation of a portion of the Internet routing at a given instant in time and to present to the user the changes in this routing over a subsequent period of time. In this section we first give a more rigorous definition of the visualization problem, that is, of the objects and events that need to be visualized. Second, we survey the visualization requirements, that is, those characteristics that, in our opinion, enhance the effectiveness of the visualization. Finally, we discuss the visualization alternatives from which we chose those that met our requirements.

6.1 The Visualization Problem

We define a *collector-peer* as a router having a BGP peering with ORV or with a RIS collector. The set of the available collector-peers is determined by the configuration of the RIS and ORV services. Each collector-peer offers a “vantage point” to investigate the routing in the Internet. From a more practical point of view, given a specific IP prefix and a specific time, for each collector-peer it is known with which AS-path it reached the prefix at that time, that is the route chosen by the collector-peer to reach the specified prefix. The last AS of the AS-path (the leftmost in the popular `show ip bgp` command output) is the one containing the collector-peer, while the first (rightmost) AS of the AS-path is the *target AS*, advertising, and hence hosting, the IP prefix.

Thus, focusing the attention on a specific prefix, we can define a *routing status* at a given time for that prefix as a collection of AS-paths, one for each collector-peer. We call *routing graph* the graph induced by the routing status, that is the graph in which each vertex is an AS occurring in at least one AS-path and there exists an edge between two vertices if the corresponding ASes appear consecutively in at least one AS-path of the routing status.

A *routing event* occurring at instant t is an update to the routing status reported by a collector-peer, and can be one of the following:

New route A collector-peer which could not reach the target prefix before time t acquires connectivity at time t by using the specified AS-path.

Route change A collector-peer which reached the target prefix using AS-path p before time t changes its routing so it uses AS-path q with $q \neq p$.

Route withdrawal A collector-peer which could reach the target prefix before time t can no longer reach it.

Further, it is sometimes desirable to consider the following routing event that is not associated with an actual routing change but is also recorded by collectors and occurs rather frequently.

Route re-announcement A collector-peer which reached the target prefix using AS-path p before time t announces at time t that it reaches it using the same AS-path p.

A *routing history* is given by a starting routing status at time t_{start} and a sequence of routing events occurring in the interval $[t_{\text{start}}, t_{\text{end}}]$. From the routing history it is possible to reconstruct the routing status at each instant in its time interval.

Thus, in more specific terms, the visualization problem addressed by BGPlay is to graphically represent the evolution of the routing status, that is its initial configuration and the events contained in the routing history.

6.2 Visualization Requirements

We distinguish between the requirements related to the visualization of a routing status and requirements related to the visualization of a routing event. For routing status visualization, we have identified the following requirements:

- The user's attention should mainly be focused on the target AS, i.e. the AS that hosts the IP prefix the user is interested in.

- An AS should appear in the drawing at a geometric distance from the target AS that is roughly proportional to the number of (ASes) hops separating them, so that the geometric distance from the AS target reflects the number of intermediate ASes needed to reach it, which may be used as a rough measure of the quality of the connectivity.
- For each visualized AS, the AS-path used to reach the target AS should be fully identifiable in the drawing, even if the path shares edges with other paths.

Observe that the last requirement would be easy to meet if the routing graph were a tree. In fact, in this case there is just one path from each AS to the target AS. The presence of cycles makes this requirement more difficult to meet.
For routing event visualization, we have identified the following requirements:

- It should be possible to convey to the user all the details of a routing history, without relying on aggregation techniques such as averaging, counting, etc.
- The visualization of a routing history may encompass the visualization of several subsequent routing statuses, whose visualization should meet the static requirements identified above for them.
- The sequence of routing statuses and routing events and the instants when they occur in the time interval should be clear to the user.
- The difference between two consecutive routing statuses, which consists of a single routing event, should be suitably pointed out.
- Conversely, the similarities between two consecutive routing statuses should be carefully exploited to maintain the user mental map [5, 16].

6.3 The Visualization Approach of BGPlay

A natural choice is to represent a routing status by using known graph drawing techniques [4] to visualize its routing graph, placing the target AS in a relevant position, such as at the center of the drawing. The next step in this approach is to choose a graph drawing standard. Our preliminary experiments showed that, although orthogonal drawings are quite compact and readable, geometrical distances between ASes are largely independent from topological distances and there is no natural way to impose the position of the target AS with respect to the other ASes (see Fig. 4).

In the current version of BGPlay we adopt the straight-line drawing standard. For such a standard there exist several layout algorithms, most of them using a force directed approach, that easily allow fixing the position of a node, and that, especially for sparse graphs, place the nodes at distances that roughly reflect their topological distances.

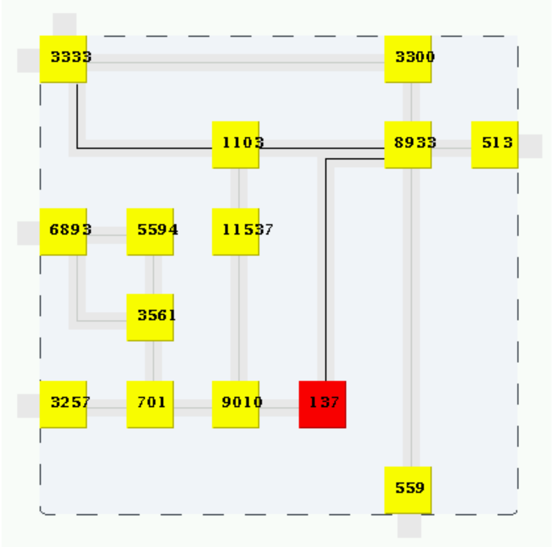


Figure 4: The first prototype of BGPlay (called `FlapViewer`) adopted an orthogonal drawing standard. The target AS (the darker node) was arbitrarily placed and the geometric distances did non reflect the topological distances. The topological distance between AS137 and AS559, for example, is two, although they are located very close in the drawing. On the other hand, AS8933 seems to be farther from AS137 than AS9010, while they have both topological distance one.

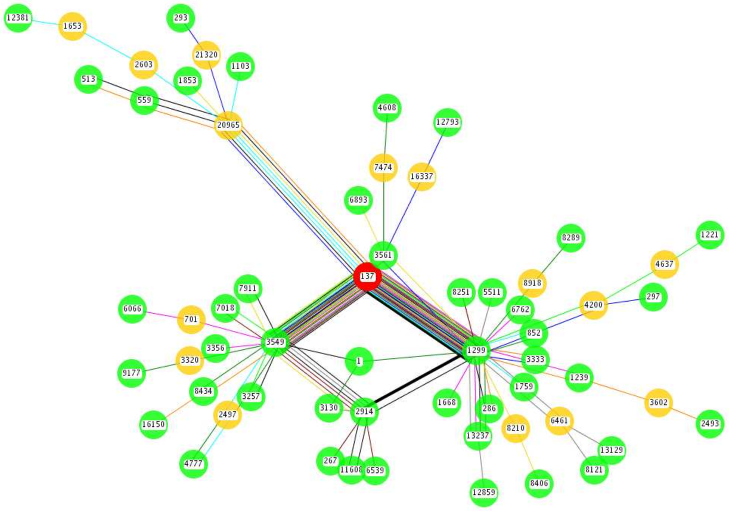


Figure 5: In a second prototype of BGPlay all AS-paths were separately displayed. From this snapshot it is apparent that paths were hardly distinguishable when they were numerous.

The third requirement for the routing status visualization is the need to fully identify each AS-path, and is much more complex to meet. Our first approach was to represent each AS-path separately and with a distinct color. A first prototype using this approach demonstrated that the number of paths to be shown was, usually, too high for the paths to be easily recognizable due to their closeness and due to the limited number of colors distinguishable on a monitor by human eyes (see Fig. 5).

Hence, we examined a different approach. As noted above, a set of AS-paths whose union is a tree can be represented without ambiguities as a tree with an arbitrary color. This is because in a tree every pair of vertices is connected by a single path, and thus there can be no doubt about the AS-path from the target AS to each AS containing a collector-peer.

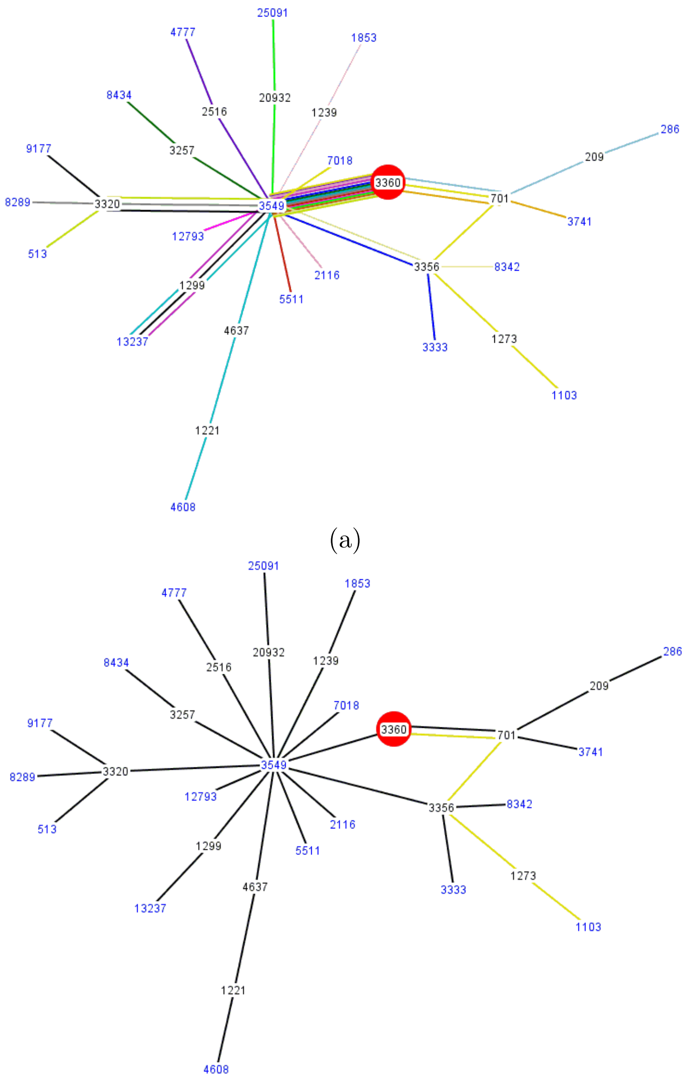
Unfortunately, the routing graph is very often not a tree. Hence, our strategy is to partition the AS-paths into sets such that the union of the AS-paths of the same set is an acyclic graph that can be unambiguously represented as a uniformly colored tree. Fig. 6 shows how this strategy produces clear and effective representations. In Section 7.1 we address the problem of partitioning the set of the AS-paths while minimizing the used colors. This technique is only applied to those AS-paths (represented dashed by BGPlay) that are not involved in any routing events. The other paths (represented solid) are assigned their colors on the basis of their collector-peers, so that paths connecting, at different instants of time, the same collector-peer with the target AS have the same color (allowing the user to perceive them as the same route which simply switched between one path and the other).

Concerning routing history visualization, the two contrasting requirements both to represent every single routing status and to convey its exact temporal position in the routing history are hard to meet within a single representation. We chose to show a time panel beside the window where the current routing status is displayed (see Fig. 10). The time panel contains a vertical time axis running from the start of the time interval (bottom) to the end of the interval (top). A cursor shows the position in the interval of the currently visualized routing status. The user may drag the cursor to jump to a different instant of time. Also, on the time axis the density of events is represented as horizontal spikes, whose length depends on the intensity of the BGP activity at that point in the time interval.

Users interested in the details of the dynamic evolution of the routing may step over consecutive routing statuses. BGPlay highlights the paths involved in the routing event between two consecutive routing statuses by means of visual effects, which are discussed in Section 7.3.

7 Visualization Algorithms

In this section we describe the main problems and algorithms related with the design and implementation of BGPlay.



(a)

(b)

Figure 6: The same routing status represented in (a) is represented in (b) by means of two differently colored trees, obtaining a clear, unambiguous, and less redundant representation.

7.1 The Tree Partition Problem

In order to minimize the number of colors needed to unambiguously represent a collection of AS-paths leading to the same target AS, following the approach described in Section 6.3, we have to solve the following problem:

Problem: TREE PARTITION

Instance: A set of AS-paths $\mathcal{P}$, such that each AS-path $p \in \mathcal{P}$ starts from a common AS_x.

Target: Find a partition for $\mathcal{P}$ in k sets such that (i) the graph induced by the AS-paths in the same set is acyclic (ii) every partition for $\mathcal{P}$ respecting (i) has at least k sets (that is, the number of sets is minimum).

Unfortunately, the TREE PARTITION problem is NP-hard. To prove this, we show the NP-hardness of the corresponding decision problem, defined as follows:

Problem: K-TREE PARTITION

Instance: A positive integer K and a set of AS-paths $\mathcal{P}$, such that each AS-path $p \in \mathcal{P}$ starts from a common AS_x.

Question: Does a partition for $\mathcal{P}$ in K sets exist such that the graph induced by the AS-paths in the same set is acyclic?

We prove that K-TREE PARTITION is NP-hard by reducing the GRAPH K-COLORABILITY to it. Recall that GRAPH K-COLORABILITY is an NP-complete problem defined as follows:

Problem: GRAPH K-COLORABILITY

Instance: A graph $G = (V, E)$ and a positive integer $K \leq |V|$.

Question: Is G K-colorable, i.e., does a coloring of the vertices of G in K colors exist such that adjacent vertices have different colors?

Theorem 1 *The problem* K-TREE PARTITION *is NP-hard.*

Proof: We reduce GRAPH K-COLORABILITY to K-TREE PARTITION. Given an instance of GRAPH K-COLORABILITY we construct the corresponding instance of K-TREE PARTITION as follows. For each vertex v of $G(V, E)$ we introduce an AS-path p_v. At the beginning of the construction all AS-paths have length one, and contain the same AS_x. Then, for each edge $(u, v) \in E$, we append to the two AS-path p_u and p_v the same $AS_{(u,v)}$. It is easy to show that the construction of the K-TREE PARTITION instance can be made in polynomial time, and that a solution for the instance of the GRAPH K-COLORABILITY problem exists iff a solution for the original instance of the K-TREE PARTITION does. □

Because of Theorem 1, BGPlay partitions the set of AS-paths by using a greedy algorithm which runs in polynomial time under the assumption that the AS-paths are bounded in length. This is guaranteed by the fact that the maximum BGP packet size of 4096 bytes limits the number of ASes in the AS-path [19]. Also, equipment from popular router manufacturers discards BGP announcements with unusually long AS-paths by default (for example, 75 ASes are considered by some equipment too many for an announcement to be valid). Finally, we note that AS-path lengths are usually shorter than the above limits suggest and that AS-paths with more than a dozen ASes are rarely seen in the Internet. In the following description of the algorithm n denotes the number of paths and the sets of AS-paths are denoted $S_0, S_1, \ldots, S_{m-1}$, where the number of sets m increases during the computation starting from 1.

1. A compatibility matrix is computed in which each AS-path is compared to every other. Two AS-paths are incompatible if and only if they form a cycle. This precomputation takes $O(n^2)$, and allows in the subsequent steps to determine if two paths are compatible in constant time.

2. Only one empty set exists at the beginning: $m = 1$ and $S_0 = \phi$

3. foreach AS-path p

    3.1 foreach set i from 0 to $m - 1$

    if p is compatible with all paths in S_i then accommodate p into S_i, skip the rest of this cycle and continue with the next path, otherwise try the next set S_{i+1} if it exists.

    3.2 if p has not been accommodated into any of the available sets add a new set S_m, initialize $S_m = \{p\}$ and increment m.

For the correctness of the algorithm it is crucial to observe that all the paths have a common endpoint (target AS). The worst case time complexity of the algorithm described above is $O(n^2)$. In fact, the number of compatibility tests performed for each AS-path is at most n and each compatibility test, thanks to the precomputed compatibility matrix, takes $O(1)$.

Fig. 7 shows a drawing produced by BGPlay using the technique described above. The target AS is 702. Three levels of gray are used. AS-paths 16150 8434 3549 702 and 1853 1239 702 are drawn in medium gray. The other paths are grouped in two trees drawn in black and light gray. Only dashed lines are used since BGPlay uses solid lines to highlight paths involved in routing changes.

7.2 The Layout

The routing graph layout is computed by using a spring embedder [4] which associates with the graph a set of bodies, one for each node, and a system of forces that can either attract or repel the nodes. Starting from an initial (pseudo-random) configuration the sum of the forces on each node is computed, and the next configuration is determined according to it. Thus, the system evolves toward an equilibrium status [4] in which nodes are "pleasantly" distributed. The forces used in BGPlay include the typical ones:

- A repulsive force is set between each pair of ASes (quadratically decreasing with their distance).

- An attractive/repulsive force is set between each pair of ASes connected by an edge, whose intensity increases linearly with the difference between the current length of the edge and its predefined "natural" length.

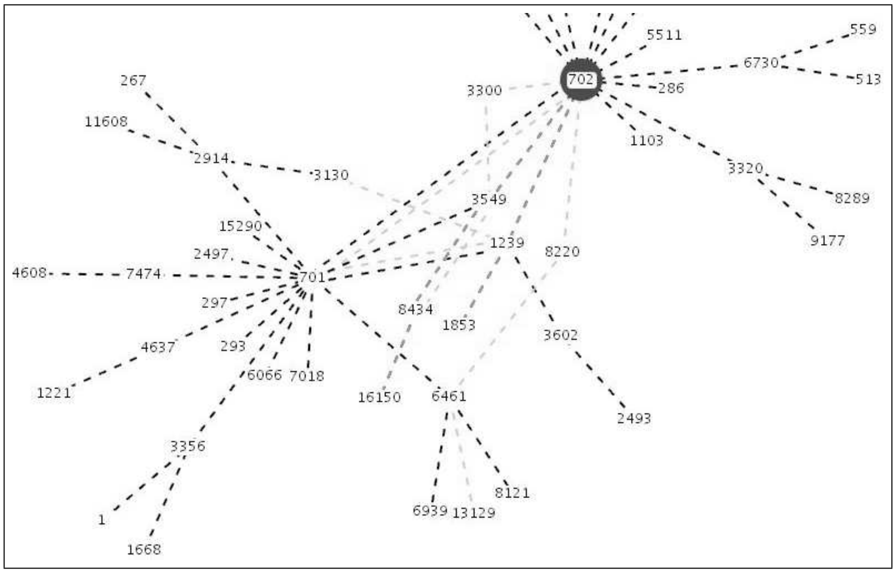


Figure 7: A drawing produced by BGPlay. Three levels of gray are used. AS-paths 16150 8434 3549 702 and 1853 1239 702 are drawn in medium gray. The other paths are grouped in two trees drawn respectively in black and light gray.

Furthermore, a specific node, corresponding to the target AS, is constrained to be in the center of the drawing area.

Also, rather than detecting when an equilibrium status is reached (a task that involves damping oscillations at the expense of the efficiency of the whole process) we used a simpler termination criterion: The number of iterations of the spring embedder algorithm is fixed, and has been empirically chosen to be high enough to reach equilibrium in all practical cases. Finally, the last 10% of the iterations involve repulsive forces between edges and nodes (where each edge is associated with twenty dummy bodies along its length). This helps to avoid overlaps between edges and nodes.

The above layout techniques may be used to produce a static drawing of a routing status. Since we need to represent a sequence of consecutive routing statuses, we compute with a single run of the above algorithm the positions of all the ASes that appear in the routing history. This position will not change during the following routing event animations (see Section 7.3). For this computation, only a subset of all the edges appearing in the routing history is considered: More specifically, only edges belonging to the initial routing graph and edges belonging to one of the first m AS paths of the routing history, where m is the minimum value sufficient to have a connected graph. Experiments considering all edges appearing in the routing history produced cluttered layouts due to the high density of the resulting graph. In fact, nodes concentrated in the center of the drawing, attracted toward each other by the high number of their adjacencies. On the other hand, a non-connected graph produces a layout where connected components are arbitrarily placed and tend to drift away one from the other. In order to reduce the number of edges while keeping the graph connected, we adopted the heuristic of neglecting those edges that appeared in some announcements only and that were not needed to connect the graph. This also favors the quality of the layout of the first routing status.

Concerning the requirements for history visualization (see Section 6.2), observe that this technique ensures that drawings of consecutive routing statuses are very similar, since node positions are unchanged. Furthermore, nodes involved in route changes (see Section 6.1) are usually placed near each other, at least for the initial part of the routing history. In fact, two AS-paths involved in a route change begin and end with the same AS, hence the spring embedder tends to draw them close together.

7.3 Graphical Features and Animation

Some graphical characteristics of BGPlay's visualization have already been introduced in the previous sections. For example, edges are drawn dashed when they refer to AS-paths which are stable during the chosen period of time and are drawn solid otherwise. Stable paths are partitioned into sets, using the heuristic algorithm described in Section 7.1, and merged into trees. Each path and each tree is drawn in a different color. An edge common to more than one tree or path is displayed using as many lines as the number of trees and paths which include that edge, where each line is drawn in the color of the corresponding tree or path. Thus, animations of routing events involve only solid paths, which are represented as polylines starting at the AS containing the collector-peer and ending at the target AS. The events fall into one of the following four categories:

New route. A new polyline corresponding to the new AS-path appears on the screen linking the nodes corresponding to the ASes in the AS-path. The thickness of the polyline pulses to attract the user's attention.

Route change. Two AS-paths are involved in this event; paths p_1 and p_2 are, respectively, the paths active before and after the routing event occurred. The polyline representing p_1 is morphed into a polyline representing p_2.

Route withdrawal. The polyline representing the AS-path pulses to attract the user's attention and then disappears.

Route re-announcement. The polyline representing the AS-path pulses to catch the user's attention but the routing status remains unchanged.

The polyline morphing used to represent a route change could be performed in several ways (see for example [1]). BGPlay employs a rather simple technique, which is illustrated in Fig. 8. Let q_1 be the polyline representing AS-path p_1 and q_2 be the polyline representing AS-path p_2. A bijection between points of q_1 and points of q_2 is defined in the following way. Polylines q_1 and q_2 are considered

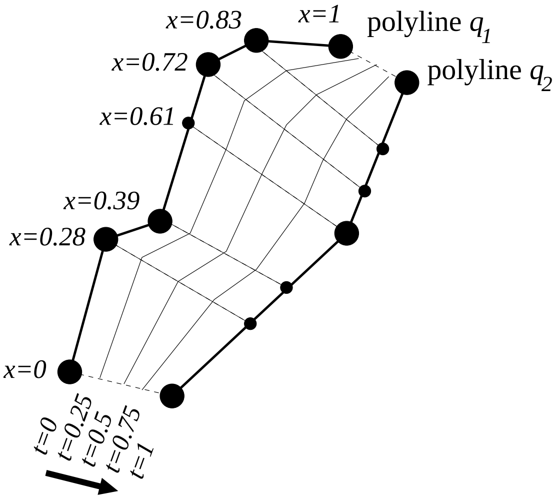


Figure 8: Route changes are animated using a linear polyline morphing.

oriented toward the target AS. Points in q_1 are mapped with real numbers x in $[0 \ldots 1]$ preserving order. The same mapping is defined for q_2. Morphing between points which have the same value of x is performed linearly both in space and in time. In our case the two extremes of a polyline are always the same, namely the target AS and the one containing the collector-peer. Fig. 8 also shows the shape of the polyline in three intermediate instants. Morphing time is set to approximately one second.

8 BGPlay Usage Examples

In this section we present the user interface of BGPlay and show how it may be used to gain greater understanding of specific routing events occurring in the Internet, both for operational and for research purposes. For the examples of this section we refer to the instance of BGPlay deployed at the RIS [2].

8.1 The BGPlay User Interface

Suppose that a user (e.g. the NOC of an ISP) is interested in examining the routing activity concerning prefix 193.0.0.0/21 during the night of May 11, 2004, because in that period some network instability was somehow perceived. The user could query the data sources directly, but the graphical nature of BGPlay makes it much easier to understand the significance of individual updates than by analyzing the updates themselves in text format (for an example, see Fig. 12).

To query BGPlay, the user connects to a web page which hosts BGPlay (in our case, the instance deployed at the RIS [2]) and starts the BGPlay applet. The BGPlay query window (Fig. 9) appears, allowing the user to specify the prefix to examine, the time interval, and the observation points to use in the query (using the RRC00, RRC01, ... checkboxes, which correspond to the Remote Route Collector managed by the RIS).

When the user submits the query, BGPlay processes the request and displays the Animation window (Fig. 10), which presents the routing information. The left part of the window contains the time panel, which plots the number of events over time. The bottom of the panel corresponds to the start of the query interval and the top of the panel to the end; the small blue triangle indicates the current time (initially, the start of the query interval). The user may jump to a specific instant within the query interval by clicking on the time panel.

The main part of the window contains the routing graph. Each number represents an AS, and the AS originating the prefix (in this case, AS 3333) is placed in the center of the graph and highlighted by a red circle. The user may obtain the name and description of an AS by clicking on it. The initial layout is performed using the techniques described in Section 7.2, but, if desired, the position of any AS in the graph may be changed by dragging it with the mouse.

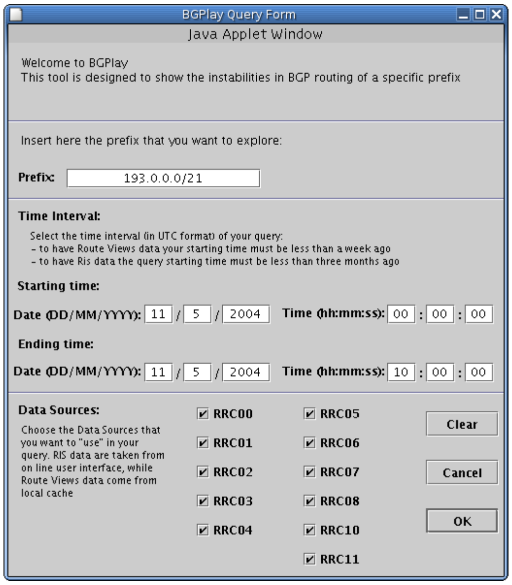
Each solid or dashed line represents a segment of an AS path seen by the data sources. A path starts in the originating AS and stops in the AS of a collector-peer; there is one path for every such peer which has an entry for the queried prefix in its routing table. The paths which did not change during the query interval are drawn dashed, while the paths which did change are drawn solid, but the color of the path itself has no special meaning: different colors are used only to ensure that each AS-path from a peer to the source AS can be unambiguously identified.

Note that the graph may contain isolated nodes which have no paths to the origin AS. This does not necessarily imply that these ASes do not have a path to the queried prefix: more usually these ASes do not contain collector-peers (and thus no information about their routing is known) and appear in the graph because they have been or will be part of a path which was in use in another moment of the query interval.

The bottom of the window contains a control panel which allows the user to toggle the display of route reannouncement events, start a new query, and move through the sequence of events that occurred in the specified time interval. Both forward and backward movement is possible. As each routing event is displayed, BGPlay updates the routing graph with a smooth animation and displays information on the event itself in the upper part of the window. This includes the event identifier, a timestamp, the type of event (see Section 7.3), the collector which recorded the event and the peer from which it was received, and any additional information appropriate to the type of event (for example, if the event is a path change, the old and new AS paths).

8.2 BGPlay for Operational Purposes

The data aggregation and visualization capabilities of BGPlay can be of great use to operators who wish to debug routing problems and to study the effects of configuration changes made in their AS on the routing used by other ASes in



Welcome to BGPlay
This tool is designed to show the instabilities in BGP routing of a specific prefix

Insert here the prefix that you want to explore:

Prefix: 193.0.0.0/21

Time Interval:

Select the time interval (in UTC format) of your query:
- to have Route Views data your starting time must be less than a week ago
- to have Ris data the query starting time must be less than three months ago

Starting time:

Date (DD/MM/YYYY): 11 / 5 / 2004 **Time (hh:mm:ss):** 00 : 00 : 00

Ending time:

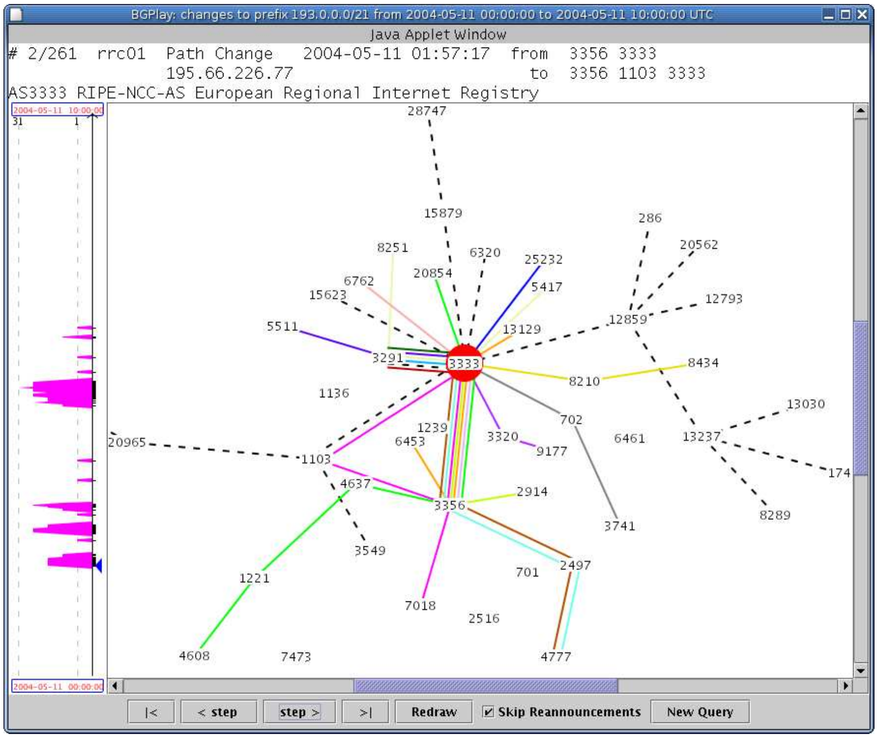
Date (DD/MM/YYYY): 11 / 5 / 2004 **Time (hh:mm:ss):** 10 : 00 : 00

Data Sources:

Choose the Data Sources that you want to "use" in your query. RIS data are taken from on line user interface, while Route Views data come from local cache

☑ **RRC00** ☑ **RRC05**
☑ **RRC01** ☑ **RRC06**
☑ **RRC02** ☑ **RRC07**
☑ **RRC03** ☑ **RRC08**
☑ **RRC04** ☑ **RRC10**
☑ **RRC11**

Figure 9: The query window



BGPlay: changes to prefix 193.0.0.0/21 from 2004-05-11 00:00:00 to 2004-05-11 10:00:00 UTC
Java Applet Window

```
# 2/261  rrc01  Path Change   2004-05-11 01:57:17  from  3356 3333
                195.66.226.77                       to  3356 1103 3333
AS3333 RIPE-NCC-AS European Regional Internet Registry
```

Figure 10: The animation window

the Internet. The following examples show how BGPlay may be employed for these purposes.

8.2.1 Example 1: Periodic Oscillations

BGPlay is very helpful in identifying time-based patterns in routing activity thanks to the time panel which plots BGP activity over time. For an example, consider Fig. 11, which shows BGPlay displaying the routing activity occurring to a particular prefix over the course of one hour. (Note that the nodes in this example were manually rearranged, grouping stable and unstable paths together in order to make the oscillations easier to see in the figures.)

The time panel clearly shows that routing activity is occurring in bursts of BGP events with a period of roughly three to four minutes, and the routing graph shows that the bursts have the effect of switching the routing between the two states shown in Fig. 11 (a) and (b) respectively. Thus we may deduce that routing to the prefix in question is subject to periodic oscillations, possibly caused by connectivity problems between the originating AS (AS28998) and one of its ISPs (AS8220), which cause the paths from many ASes to the target AS to switch periodically back and forth from using AS8220 to using AS28998's other ISP, AS3243.

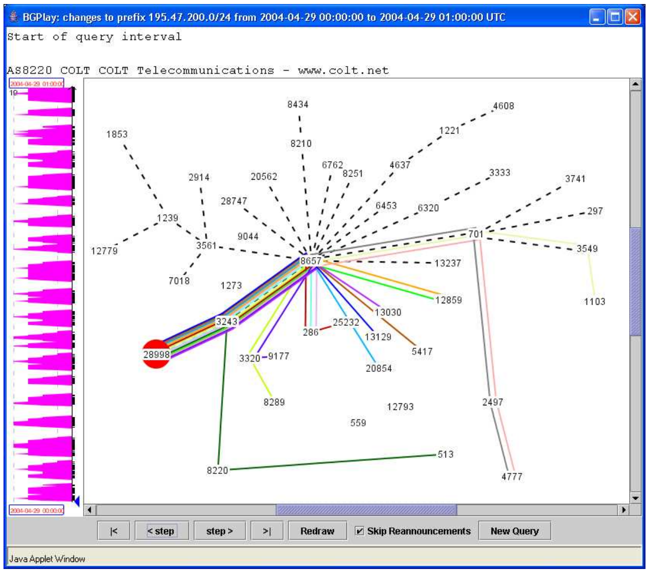
Compare this representation to the list of over 1200 updates returned by the RIS web interface for the same time period, of which a small sample is shown in Fig. 12 (the full list of updates is over 80 screenfuls long). Clearly, the visual representation provided by BGPlay is much easier to understand and can thus provide greater insight into the cause of the problem.

8.2.2 Example 2: Examining the Effects of a New Upstream

In Fig. 13, BGPlay shows the routing events that took place when AS3313 added AS3257, previously a peer, to its existing upstreams AS1299, AS5400 and AS15589. AS3257 therefore started providing transit to AS3313's prefixes. The graph clearly shows BGP converging into a new state in which most of the collector-peers reach AS3313 through the new provider and some continue to use the existing providers. The operators of AS3313 might then have evaluated the impact of the new upstream and whether it had the desired effect on AS3313's connectivity simply by looking at the graph. Thus, BGPlay allows network operators to understand quickly and easily how local changes in their routing policy and configuration affect the AS's global connectivity.

8.2.3 Example 3: Use of BGPlay for Traffic Engineering

BGPlay can be used as a feedback mechanism for fine-tuning routing policy. For an example, consider Fig. 14. In this example, the operators of AS559 (SWITCH) decided to add AS prepending on the routes announced to one of their upstream providers, AS3549, so as to improve load balancing between their two upstreams AS1299 and AS3549. BGPlay clearly shows that some of

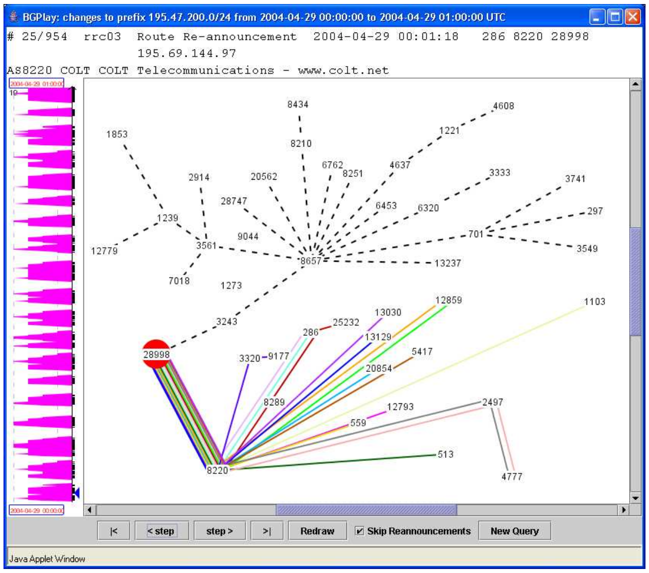


BGPlay: changes to prefix 195.47.200.0/24 from 2004-04-29 00:00:00 to 2004-04-29 01:00:00 UTC

```
Start of query interval

AS8220 COLT COLT Telecommunications - www.colt.net
```

(a)

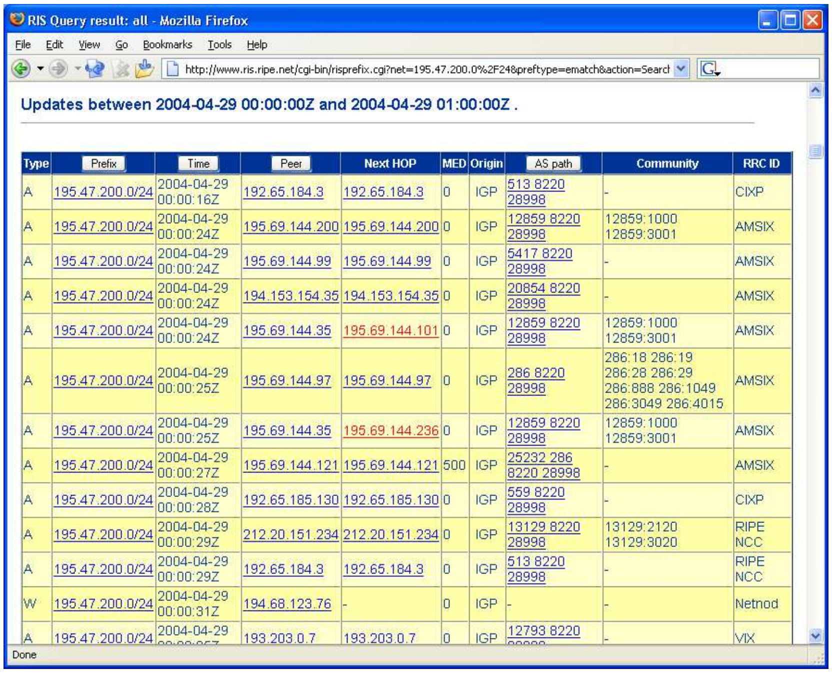


BGPlay: changes to prefix 195.47.200.0/24 from 2004-04-29 00:00:00 to 2004-04-29 01:00:00 UTC

```
# 25/954  rrc03  Route Re-announcement  2004-04-29 00:01:18   286 8220 28998
                 195.69.144.97
AS8220 COLT COLT Telecommunications - www.colt.net
```

(b)

Figure 11: A prefix in AS28998 periodically oscillating between two routing states: (a) all collector peers see the prefix through one of its ISPs, AS3243; (b) some peers change their paths to the prefix so that they go through another of AS28998's ISPs, AS8220. The time panel shows bursts of BGP activity occurring approximately every three to four minutes. Examination of the routing graph suggests that these may be caused by a connectivity problem between AS28998 and AS8220.


RIS Query result: all - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Done

http://www.ris.ripe.net/cgi-bin/risprefix.cgi?net=195.47.200.0%2F24&preftype=ematch&action=Searc

Updates between 2004-04-29 00:00:00Z and 2004-04-29 01:00:00Z .

Type	Prefix	Time	Peer	Next HOP	MED	Origin	AS path	Community	RRC ID
A	195.47.200.0/24	2004-04-29 00:00:16Z	192.65.184.3	192.65.184.3	0	IGP	513 8220 28998	-	CIXP
A	195.47.200.0/24	2004-04-29 00:00:24Z	195.69.144.200	195.69.144.200	0	IGP	12859 8220 28998	12859:1000 12859:3001	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:24Z	195.69.144.99	195.69.144.99	0	IGP	5417 8220 28998	-	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:24Z	194.153.154.35	194.153.154.35	0	IGP	20854 8220 28998	-	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:24Z	195.69.144.35	195.69.144.101	0	IGP	12859 8220 28998	12859:1000 12859:3001	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:25Z	195.69.144.97	195.69.144.97	0	IGP	286 8220 28998	286:18 286:19 286:28 286:29 286:888 286:1049 286:3049 286:4015	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:25Z	195.69.144.35	195.69.144.236	0	IGP	12859 8220 28998	12859:1000 12859:3001	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:27Z	195.69.144.121	195.69.144.121	500	IGP	25232 286 8220 28998	-	AMSIX
A	195.47.200.0/24	2004-04-29 00:00:28Z	192.65.185.130	192.65.185.130	0	IGP	559 8220 28998	-	CIXP
A	195.47.200.0/24	2004-04-29 00:00:29Z	212.20.151.234	212.20.151.234	0	IGP	13129 8220 28998	13129:2120 13129:3020	RIPE NCC
A	195.47.200.0/24	2004-04-29 00:00:29Z	192.65.184.3	192.65.184.3	0	IGP	513 8220 28998	-	RIPE NCC
W	195.47.200.0/24	2004-04-29 00:00:31Z	194.68.123.76	-	0	IGP	-	-	Netnod
A	195.47.200.0/24	2004-04-29	193.203.0.7	193.203.0.7	0	IGP	12793 8220	-	VIX

Figure 12: Output returned by the RIS web interface for the routing events displayed by BGPlay in Fig. 11. The full list of updates for one hour of oscillations is over 80 screenfuls long and contains over 1200 updates.

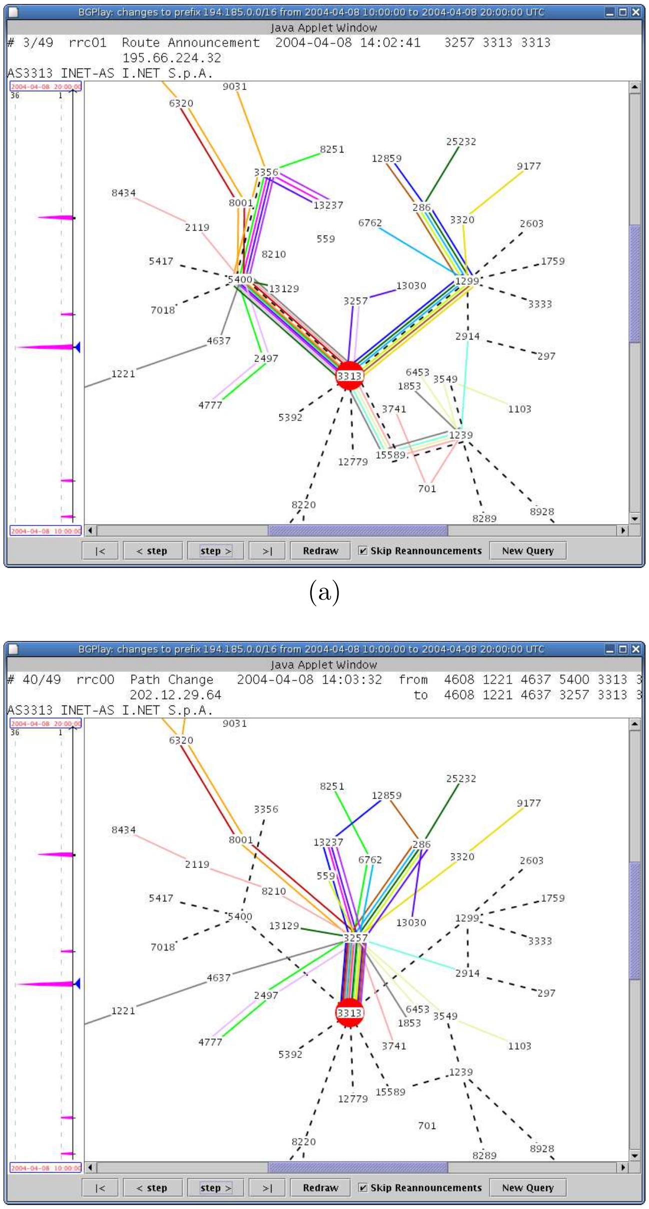
the ASes which previously reached SWITCH through AS3549 switched to using AS1299 as a result of the change. Using BGPlay, the operators of AS559 were able to verify in near real-time whether the configuration change had had the desired effect.

8.3 BGPlay for Research Purposes

The graphical nature of BGPlay can be of great help in understanding routing events. The following examples show how BGPlay may be employed for research purposes and teaching.

8.3.1 Example 1: Announcing a New Prefix

BGPlay can also help understand the propagation of BGP announcements and the BGP convergence progress. Fig. 15 gives only a hint of what can be observed by asking BGPlay to display the routing events that occur when a new prefix is announced for the first time. Studying such an event can provide insight into how announcements are propagated and into the BGP convergence process. By using BGPlay it can be noticed that the announcement of a single prefix, even in the absence of other factors, leads to more complex behavior than the simple



BGPlay: changes to prefix 194.185.0.0/16 from 2004-04-08 10:00:00 to 2004-04-08 20:00:00 UTC

```
# 3/49  rrc01  Route Announcement  2004-04-08 14:02:41   3257 3313 3313
               195.66.224.32
AS3313 INET-AS I.NET S.p.A.
```

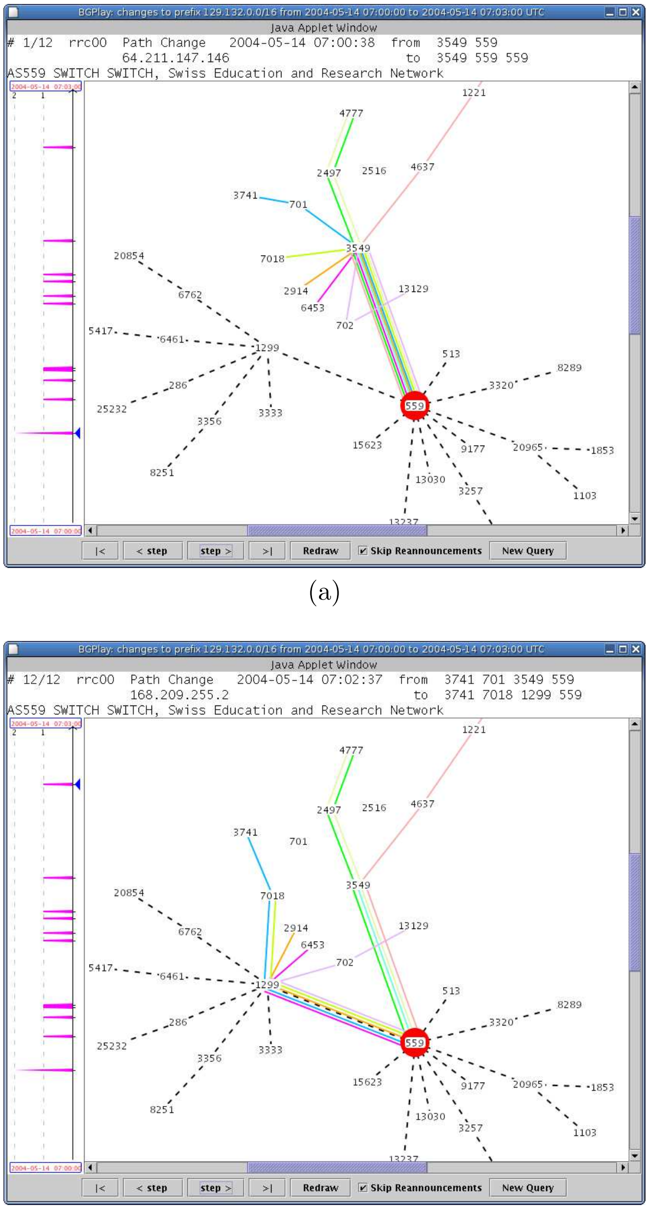
(a)

BGPlay: changes to prefix 194.185.0.0/16 from 2004-04-08 10:00:00 to 2004-04-08 20:00:00 UTC

```
# 40/49  rrc00  Path Change   2004-04-08 14:03:32  from  4608 1221 4637 5400 3313 3
                202.12.29.64                          to  4608 1221 4637 3257 3313 3
AS3313 INET-AS I.NET S.p.A.
```

(b)

Figure 13: AS3257 starts behaving as a transit AS for the prefixes originated by AS3313: (a) before the new policy takes effect; (b) after AS3257 propagates AS3313 announcements and BGP converges. Many of the collector peers have switched paths from using the existing upstream providers (AS1299, AS5400 and AS15589) to AS3257. This information can be of great use to the operators of AS3313, as it allows them to examine the effect of the new peering on the AS's global connectivity in a quick and easy way.



```
BGPlay: changes to prefix 129.132.0.0/16 from 2004-05-14 07:00:00 to 2004-05-14 07:03:00 UTC
Java Applet Window
# 1/12  rrc00  Path Change   2004-05-14 07:00:38  from  3549 559
               64.211.147.146                       to  3549 559 559
AS559 SWITCH SWITCH, Swiss Education and Research Network
```

(a)

```
BGPlay: changes to prefix 129.132.0.0/16 from 2004-05-14 07:00:00 to 2004-05-14 07:03:00 UTC
Java Applet Window
# 12/12  rrc00  Path Change   2004-05-14 07:02:37  from  3741 701 3549 559
                168.209.255.2                        to  3741 7018 1299 559
AS559 SWITCH SWITCH, Swiss Education and Research Network
```

(b)

Figure 14: Studying the effects of traffic engineering. AS559 (SWITCH) changes the AS prepending on the routes it announces to one of its upstreams to influence other ASes's routing toward itself: (a) before the change; (b) after the change. Note that a number of collector peers have switched paths and now use AS1299 instead of AS3549 to reach AS559. Using BGPlay the operators of AS559 were able to see in near real-time whether the configuration change had had the desired effect.

propagation of the announcement itself. For example, differences in update propagation times, router load, and BGP announcement timers lead to updates arriving out of order and thus, in addition to the expected announcements, to path changes as well.

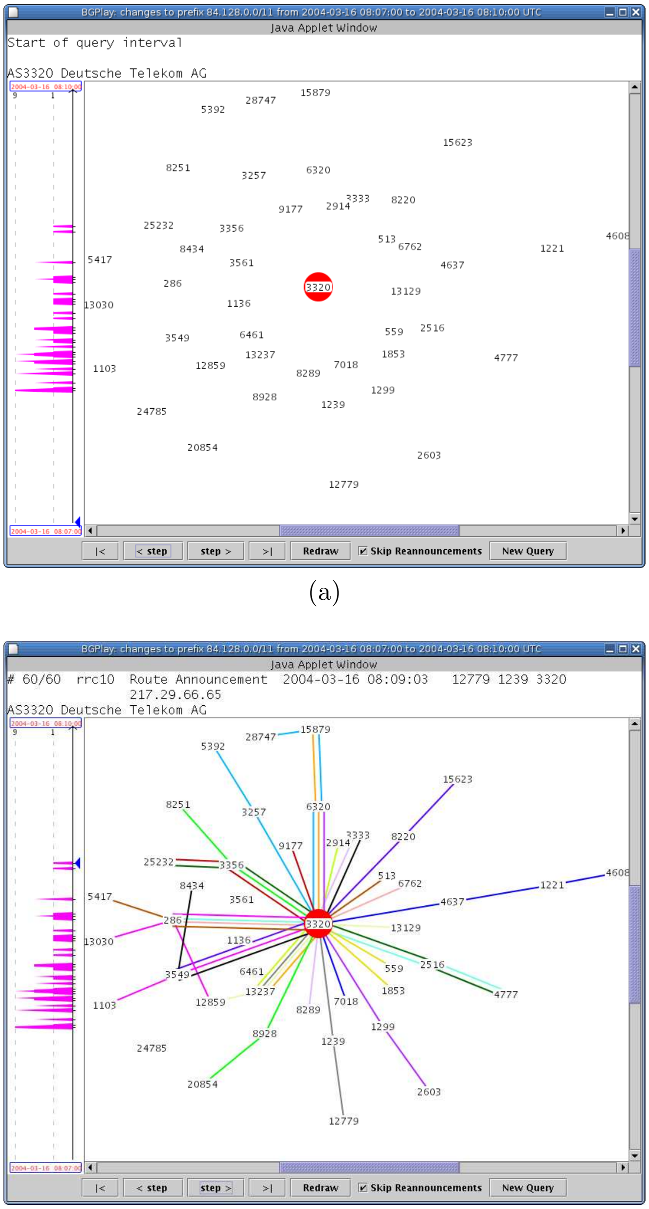
8.3.2 Example 2: a Routing Beacon

BGP beacons [15] are BGP prefixes which are announced and withdrawn at known times for research purposes. They are used to study the convergence properties of BGP and allow researchers to study in detail the sequence of events that occur when a route is announced or withdrawn; BGPlay, thanks to its aggregation of multiple data sources and its visual representation, greatly aids the comprehension of such events. An example of a BGP beacon is in Figs. 16, 17, and 18 (note that the nodes in this example were manually rearranged to clarify the situation near the origin AS). The routing activity of the beacon is similar to that of the example in Fig. 11 in that there are bursts of BGP activity between two stable states; however, as can be seen from the time panel, the bursts of BGP events are more regular and more equally spaced over time due to the fact that the beacon is programmed to announce and withdraw routes exactly every two hours.

Studying BGP beacons with BGPlay can help understand the basic behavior of BGP in simple topologies and controlled conditions. This is especially interesting in the case of withdrawals, which, due to the BGP's path-vector nature, do not propagate immediately but entail a period in which BGP explores alternate, usually longer, paths before concluding that a destination is in fact unreachable [13]. In the example of Fig. 17 the alternate paths to the withdrawn destination are all longer than two.

Acknowledgments

We would like to thank Simon Leinen from SWITCH for providing valuable feedback and real-world examples of the use of BGPlay by network operators. Also, we would like to thank Henk Uijterwaal for providing us with the opportunity to host an instance of BGPlay at the RIPE NCC and Randy Bush and the RouteViews team for the opportunity to host an instance of BGPlay at Oregon University.



BGPlay: changes to prefix 84.128.0.0/11 from 2004-03-16 08:07:00 to 2004-03-16 08:10:00 UTC

Java Applet Window

Start of query interval

AS3320 Deutsche Telekom AG

(a)

BGPlay: changes to prefix 84.128.0.0/11 from 2004-03-16 08:07:00 to 2004-03-16 08:10:00 UTC

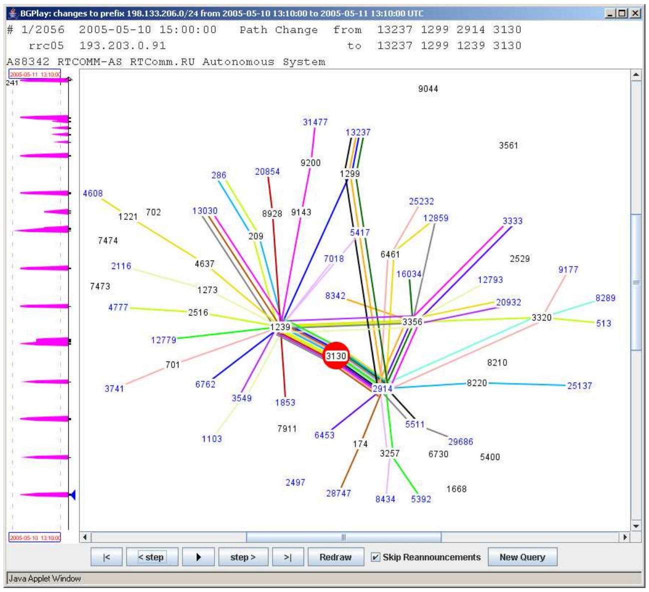
Java Applet Window

60/60 rrc10 Route Announcement 2004-03-16 08:09:03 12779 1239 3320
217.29.66.65

AS3320 Deutsche Telekom AG

(b)

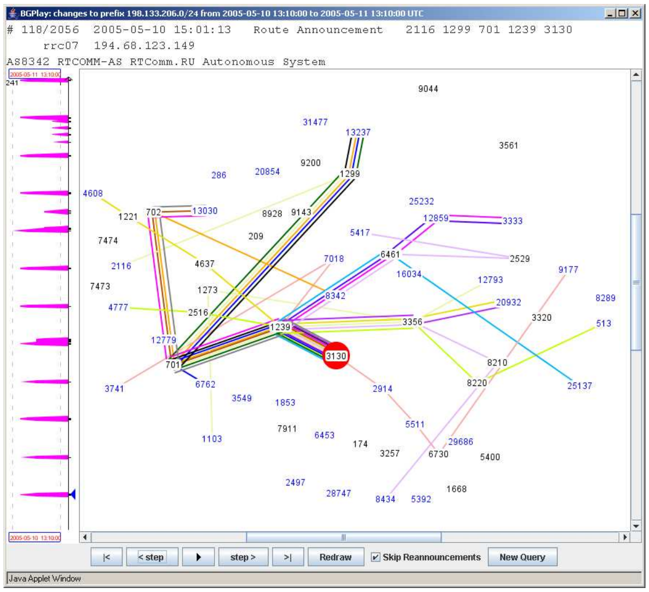
Figure 15: The new prefix 84.128.0.0/11 is announced at about 2004-03-16 08:08:00 UTC: (a) before the announcement; (b) after convergence. Studying the order of routing events can provide insight into the way in which timing factors influence the propagation of announcements and can lead to path changes during convergence.



BGPlay: changes to prefix 198.133.206.0/24 from 2005-05-10 13:10:00 to 2005-05-11 13:10:00 UTC

```
# 1/2056  2005-05-10 15:00:00   Path Change  from  13237 1299 2914 3130
   rrc05  193.203.0.91                         to  13237 1299 1239 3130
AS8342 RTCOMM-AS RTComm.RU Autonomous System
```

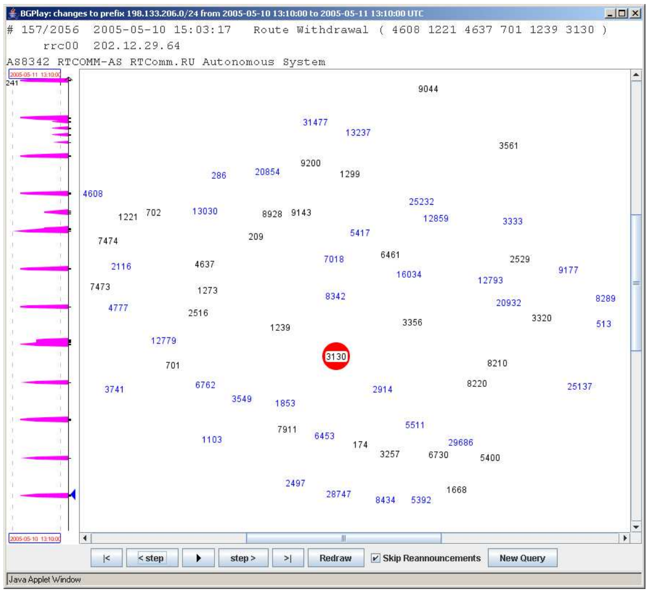
Figure 16: A BGP beacon periodically announcing and withdrawing the prefix 198.133.206.0/24 from 2005-05-10 13:10:00 UTC to 2005-05-11 13:10:00 UTC. Note the regular patterns of activity in the time panel which correspond to the beacon being announced and withdrawn with a period of exactly two hours. The picture shows the routing status just before a withdrawal.



BGPlay: changes to prefix 198.133.206.0/24 from 2005-05-10 13:10:00 to 2005-05-11 13:10:00 UTC

```
# 118/2056  2005-05-10 15:01:13   Route Announcement   2116 1299 701 1239 3130
     rrc07  194.68.123.149
AS8342 RTCOMM-AS RTComm.RU Autonomous System
```

Figure 17: A routing status following that of Fig. 16 where shorter AS-paths disappeared due to the propagation of the withdrawal.



BGPlay: changes to prefix 198.133.206.0/24 from 2005-05-10 13:10:00 to 2005-05-11 13:10:00 UTC

```
# 157/2056  2005-05-10 15:03:17   Route Withdrawal ( 4608 1221 4637 701 1239 3130 )
     rrc00  202.12.29.64
AS8342 RTCOMM-AS RTComm.RU Autonomous System
```

Figure 18: The routing status after the complete propagation of the withdrawal of Figs. 16 and 17.

References

[1] S. Bespamyatnikh. An optimal morphing between polylines. *International Journal of Computational Geometry & Applications*, 12(3):217–228, 2002.

[2] BGPlay instance hosted by the RIS project [Online]. `http://www.ris.ripe.net/bgplay/`.

[3] A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the high level structure of the Internet with Hermes. *J. of Graph Algorithms and Applications*, 6(3):281–311, 2002.

[4] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[5] P. Eades, R. F. Cohen, and M. L. Huang. Online animated graph drawing for Web navigation. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 330–335. Springer-Verlag, 1997.

[6] L. Gao and J. Rexford. Stable Internet routing without global coordination. In *Measurement and Modeling of Computer Systems*, pages 307–317, 2000.

[7] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *IEEE INFOCOM 2000*, pages 1371–1380, Tel Aviv, Israel, March 2000.

[8] T. Griffin and G. T. Wilfong. An analysis of BGP convergence properties. In *SIGCOMM*, pages 277–288, 1999.

[9] Hermes [Online]. `http://www.dia.uniroma3.it/~hermes/`.

[10] B. Huffaker, D. Plummer, D. Moore, and k claffy. Topology discovery by active probing. Technical report, Cooperative Association for Internet Data Analysis - CAIDA, San Diego Supercomputer Center, University of California, San Diego, 2002.

[11] G. Huston. Interconnection, peering and settlements – Part 1. *Internet Protocol Journal*, 2(1):2–16, 1999.

[12] G. Huston. Interconnection, peering and settlements – Part 2. *Internet Protocol Journal*, 2(2):2–23, 1999.

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

[14] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, 1998.

[15] Z. M. Mao, R. Bush, T. Griffin, and M. Roughan. BGP beacons. In *IMC 2003*, Oct. 2003.

[16] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Lang. Comput.*, 6(2):183–210, 1995.

[17] Multithreaded Routing Toolkit (MRT) Project [Online]. `http://www.mrtd.net`.

[18] University of Oregon RouteViews project [Online]. `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) [Online]. `http://www.ripe.net/ripencc/pub-services/np/ris/`.

[21] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proceedings of ACM/SIGCOMM '02*, Aug. 2002.

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

[23] A. S. Tanenbaum. *Computer Networks.* Prentice-hall International, Inc., 1996. ISBN: 0-13-394248-1.