Realistic BGP Traffic for Test Labs

Olaf Maennel and Anja Feldmann
Saarland University, Saarbrücken
{olafm,anja}@net.uni-sb.de

ABSTRACT

This paper examines the possibility of generating realistic routing tables of arbitrary size along with realistic BGP updates of arbitrary frequencies via an automated tool deployable in a small-scale test lab. Such a tool provides the necessary foundations to study such questions as: the limits of BGP scalability, the reasons behind routing instability, and the extent to which routing instability influences the forwarding performance of a router.

We find that the answer is affirmative. In this paper we identify important characteristics/metrics of routing tables and updates which provide the foundation of the proposed BGP workload model. Based on the insights of an extensive characterization of BGP traffic according to such metrics as prefix length distributions, fanout, amount of nesting of routing table prefixes, AS path length, number and times between BGP update bursts and number and times between BGP session resets, etc., we introduce our prototype tool, `RTG`. `RTG` realizes the workload model and is capable of generating realistic BGP traffic. Through its flexibility and parameterization `RTG` enables us to study the sensibilities of test systems in a repeatable and consistent manner while still providing the possibility of capturing the different characteristics from different vantage points in the network.

Categories and Subject Descriptors

C.2.2 [**Computer Communication Networks**]: Routing Protocols

General Terms

Measurement, Design, Performance

Keywords

BGP, Workload

SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

1. INTRODUCTION

New features (e.g., fair queuing), new components (e.g., new router architectures, new software versions), are incorporated into the Internet every day. Ideally each component should be tested for its correctness and evaluated for its effectiveness in a test environment before it is deployed in the network. Unfortunately the ability to test many features is limited by the simplicity of current test setups. Typical testbeds consist of a small number of routers and test-traffic generators, e.g., IXIA [1] from Ixiacom, Chariot [2] from NetIQ, TeraRouter Tester [3] from Spirent Communication and RIG [4] from Arsin. Some [2] are only capable of generating test traffic, others [1, 3, 4] can also generate routing protocol traffic, but all of them suffer a severe shortcoming: the traffic they generate is not necessarily consistent with the traffic in the Internet. For example test-traffic often does not reflect temporal variability as captured by self-similarity nor does it capture the full range of IP addresses. While some commercial tools support routing protocols their abilities are limited to the basic operations: propagation of simple updates and participation in the exchange of routing tables or synchronization of topology databases. This is at least partly due to our limited understanding of the dynamics of routing protocols, e.g., [5, 6, 7]. Therefore it is currently impossible to, e.g., recreate complex but realistic BGP (Border Gateway Protocol [8, 9, 10]) routing instabilities as observed in the Internet in a test-lab, except by replaying a captured trace.

BGP controls the routing between autonomous systems (AS). Recent work by researchers [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 5], and the standardization body, IETF (especially the working groups on Inter Domain Routing (idr) [21, 22] and on Benchmarking Methodology (bmwg) [23, 24]) have shown that BGP's dynamics are poorly understood. On the other hand for network operators it is crucial to understand BGP's dynamics, as can be seen by the numerous presentations and panels on BGP at the network operator forums, e.g., NANOG [25]. In this work we set out to

- identify a structure in BGP traffic
- characterize the structure using actual measurements
- exploit the structure for a BGP workload model
- propose a tool, `RTG`, to realize the workload model
- and show, in parts, that `RTG` can create realistic BGP traffic.

In short the goal of this paper is two-fold: first to identify and characterize BGP traffic and therefore contribute to the basic understanding of BGP dynamics and second to repro-

duce realistic BGP traffic, using our tool RTG, in a small test-lab consisting of only a few physical components.

Such a tool will allow us to explore BGP in many different ways: establishing basic BGP implementation regression tests, testing BGP implementation features, finding good settings for the many BGP parameters (especially the BGP timers), testing BGP's scalability, testing the interactions of BGP's routing table updates with packet forwarding, experimenting with changes in the BGP workload (e.g., different vantage points or changes to the protocol), understanding the reasons for BGP instability, etc. But RTG is not just useful for understanding BGP traffic, its scalable routing table models are also useful for packet lookup and classification algorithm designers.

RTGs approach differs from black-box approaches, that just replay a trace, in that it captures the structure of BGP updates and their imposed changes. It is highly configurable, parameterizable and scalable and therefore allows insights into the reasons behind certain behaviors. In contrast our ability to scale a trace and/or adapting it to a different scenario is rather limited. In addition traces are usually considered proprietary. But tools, such as RTG, can be used by people at different locations and companies, that normally could not share data to normalize their workloads [26]. But maybe most importantly the tool highlights how well we understand the measured data. Can we capture the essential pieces in a workload model so that we can reproduce reality based on a modeled structure?

Our model-based BGP reference stream is designed around the notion of a **workload model** that, we believe, captures the structure of BGP traffic in a similar way as other workload models, such as SURGE [27] or tcplib [28, 29], capture the structure and characteristics of Web or TCP traffic. In this paper we

- explain the characteristics we decided to include in the model and our reasoning for including them (Section 3).
- present a novel characterization of BGP traffic following the elements of the workload model (Section 4).
- discuss how the workload model is used by RTG to generate BGP traffic (Section 5).
- demonstrate with examples that RTG generates realistic BGP traffic (Section 6).
- summarize our experience and suggest future research directions (Section 7).

Note that RTGs configuration files can either be automatically generated via BGP traffic analysis or manually derived or any combination of the two methods.

2. BGP BACKGROUND

The Internet is divided into a collection of autonomous systems. Routing through the Internet is accomplished on a prefix by prefix basis and depends on protocols for routing within individual ASes, e.g., EIGRP, OSPF, IS-IS, and RIP [30] and for routing between ASes [8, 31], for which BGP [9], a path-vector protocol, is the de facto standard. BGP advertisements are exchanged over BGP sessions between pairs of routers.

Upon startup the routing tables and the forwarding tables of a router need to be initialized. For BGP this means that BGP sessions to all peers of the routers have to be established. Once a BGP session to a peer is established, the two peers have to exchange their BGP routing tables. This is done by sending BGP updates for each prefix in the routing tables. Each router receiving a BGP announcement applies some local policy regarding accepting the update, adds the update to its BGP routing table (possibly replacing an already existing route for this prefix-peer-combination) and, if the "best" route changes, updates the forwarding table. In a next step, the router applies its outbound policy to the new best route and, after potentially rewriting some attributes, sends the update to its other peers. Each router receiving a BGP withdraw, deletes the entry for this prefix from the peer's BGP table. It possibly calculates a new best route to replace the withdrawn route, forwards the update to the other peers and updates its forwarding table. If no announcement or withdraw is sent for a specific time period BGP uses keepalive messages to determine if the session is up or down. If a router notices a session as *down* the corresponding routes have to be deleted from the table and updates, either announcements of alternative routes or withdraws, have to be sent to the other BGP peers.

BGP updates are limited by timers: e.g., the *Min-Route Advertisement Interval timer* [31] limits the number of updates for each prefix/session for each peer to one every x seconds. (A typical value for x is 30 seconds.) Routing updates that flow through a network can cause other updates to be generated, e.g., consider the scenario shown in Figure 1. AS1 has added a prefix P and is therefore sending a BGP update to AS2 and AS3 for P with AS path: AS1. This update is received by AS2, added to the routing tables, and sent to AS4, since AS2 has not sent an update to AS4 within the last 30 seconds. AS4 receives the update, adds the prefix to its routing table and forwards the update to AS5. AS3 also receives the update and adds it to its routing table, but instead of sending the update immediately to AS4, AS3 has to wait until the Min-Route Advertisement Interval timer expires. Once AS4 receives the update from AS3 it realizes that this is a better path, reannounces its routing table entry for this prefix and sends another update for prefix P to AS5. In this rather simple example AS1 originated one update for prefix P, yet AS4 is originating two updates for the same prefix. More general a single update originated by some AS can cause a sequence of updates, called *update sequence*, to be observed at some other AS in the Internet.

To further limit the number of updates *route flap damping* [32] has been introduced. Route flaps can be caused by administrative changes, such as additions and removals of network interfaces and network links, or administrative changes to BGP session characteristics, or due to session resets due to link failures or transport layer connectivity failures and other scenarios [22]. Since flaps can generate update sequences that propagate through the Internet, consume router resources, and may cause other routing updates, one wants to limit the scope of route flap propagation via route flap damping. The idea behind route flap damping is to use the history of updates associated with a prefix to predict its future behavior and in this way suppress oscillating routes until they have stabilized. Typical values for route flap damping, according to the recommendations from RIPE [33], are suppression periods of $30-60$ $(10-30)$ minutes for /22 to /32 (all others) after the 4th change. Other parameter settings impose damping in a progressive fashion: the more flaps the longer the suppression times.

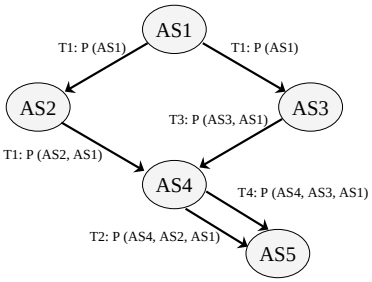

Fig. 1: Update propagation.

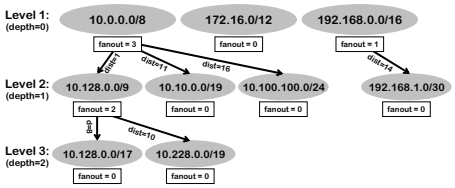

Fig. 2: Example: prefix vs. forest.

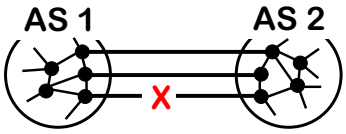

Fig. 3: Example: Fraction of prefixes updates in session reset.

3. BGP WORKLOAD INGREDIENTS

The goal of this Section is to identify essential components of BGP traffic that will help us build a workload model for BGP traffic. With BGP dynamics one typically refers to a series of updates caused by routing changes. Accordingly we need a notion that captures the cause of routing instabilities and their effects: the BGP convergence process [18, 34, 35] that they create and the changes in the BGP table that they impose. To understand the changes we need a baseline that captures the basic structure/hierarchy of the prefixes and their attributes[1] in the BGP table. In general the set of AS path attributes reflects the interconnectivity of the various ASes and their peering policies. But from the viewpoint of a single BGP peering session the AS path is the ingredient that captures correlations between routing updates for different prefixes as well as for the same prefix. Other attributes such as MED and communities reflect the external policy of the peering AS.

We propose to capture the cause of a routing instability by the notion of an *instability creator* and the temporal characteristic of the resulting sequence of updates by the notion of an *instability burst*. To capture the BGP table we propose to view the set of prefixes as nodes in a graph, where edges capture the structure (the nesting) of the prefixes. Correspondingly we propose to view the prefixes in the BGP table via a *prefix forest*. Instead of modeling the AS topology [36, 37] and their peering policies [38, 39, 40] explicitly we propose to focus on *AS path properties* as seen via a single peering session. After all we are looking for a workload model that can stimulate a system under test/router or a simulation but not a full BGP simulation/emulation as for example provided by SSFNet [41]. To capture the correlations within an instability burst we focus on the *attribute changes* between updates for the same prefix.

Instability creator

Routing instabilities can be caused by an AS or a prefix in one of the following ways:

- BGP session establishment/teardown/reset
- BGP session parameter change, including local filtering policy changes as well as misconfigurations
- link failure/repair
- addition/deletion of network prefixes
- prefix policy changes.

In the first case it is easy to identify the instability creator/creators. The **two peering ASes** are the instability creators if an external BGP (EBGP) session is struck. In case of an internal BGP (IBGP) session two things may happen: a sizeable set of prefixes experiences attribute changes and these changes are propagated to EBGP or a small number of prefixes suffer from the effects. This depends on the internal structure of the network and the IBGP/IGP (Interior Gateway Protocol) configuration. For backbones of tier-1 providers the latter should affect no[2] prefix or a small number of prefixes[3]. If a sizeable fraction of the prefixes are affected the **session AS** is the creator, otherwise we treat the updates as prefix additions/deletions or prefix policy changes discussed later.

The second case equals the first one since historically changes to the BGP session parameters and the local filtering policies did not take effect until after the BGP session has been cleared using a hard reset. Using soft resets it is not necessary to tear down the BGP session. Yet if the parameter changes affect a sizeable set of the prefixes using this peering session, a sizeable set of updates will be created by the two peering ASes or the single AS. Therefore this case is not distinguishable from the previous one and the **two peering ASes** are or the **AS** or the **prefixes** are considered to be the instability creators. If only a small set of the prefixes is affected we treat the created instabilities as prefix policy changes.

A link failure/repair of a peering link more or less implies that the corresponding BGP session is torn down/re-established. Therefore we do not need to consider this case separately. A link failure/repair of a backbone link is different. It does not cause an IBGP session reset but might change some IGP path cost and therefore it can create some number of updates. But most of the time only a small set of the prefixes is affected and we again treat this as a prefix policy change. If an access link fails only prefixes connected via the access link will be affected. Most of the time this will be a small number and therefore we can again tread this as prefix addition/deletion or policy change.

In the last two cases the instability creator is **the prefix** rather than any AS. The distinction between these two cases is that additions and deletions of network prefixes occur only at the originating AS, while policy changes, e.g., changes to the AS path, the communities, etc., can happen anywhere along the AS path.

In summary, we consider an AS or two ASes as the instability creator if some sizeable fraction of the prefixes using this AS are involved in updates within a reasonable short time period. Otherwise the prefix itself is responsible for the instability. In this way we are able to capture the cor-

[1]Typical BGP attributes are the AS path, MED, communities, etc.

[2]For example if IBGP with route reflectors is run on all backbone routers and each router is peering with at least two route reflectors.

[3]For example with a fully meshed IBGP configuration.

updates	interarrival time
	attributes changes
update	interarrival time
bursts	duration
	# of updates
possible	interarrival time
session	duration
resets	# of updates
	# of prefixes

Tab. 1: Metrics: BGP updates.

routes within the IP address range
prefix length
depth
fanout
distance

Tab. 2: Metrics: Routing table.

of originating routes
of transiting routes
AS path length
of unique ASes on AS path
of duplicate ASes on AS path
position duplicate ASes on AS path
distance of ASes to peer

Tab. 3: Metrics: Attributes.

relations between updates for different BGP prefixes. Note that human misconfigurations of BGP [43] can be expressed either as instabilities caused by an AS or by a prefix. This only depends on the number of affected prefixes.

Instability bursts

An instability creator may generate several *instability events*. For example a prefix instability creator associated with a flapping link to a single homed customer would create a withdraw, followed by an announce, followed by a withdraw, followed by an announce, etc., while an AS instability creator associated with a session reset to a single homed customer AS will create instability events for all prefixes originated by the AS. Each instability event is an update which may or may not be observable in the measured data. Only if the AS hop distance is one we can expect to see all instability events. The larger the distance between the creator and the observer is the more likely it is that an intermediary will have an alternative way of reaching the prefix. Accordingly the intermediary may or may not relay the original instability event and the related BGP updates. Therefore the observer will note a chain of updates, called an update sequence, that are triggered by the original instability event. Each observed update sequence captures the set of updates created by the route convergence process and may not contain the original instability event update. We refer to the total resulting set of updates as an **update burst**.

Ideally we would like to build our workload model around instability event updates and update sequences. Unfortunately distinguishing between instability event updates and related updates is an unresolved problem [7]. But a second look reveals that to a system under test it does not matter if the update burst is the result of n or m instability events. What matters is the number of updates it has to handle and the relationship between the updates. Therefore we propose to build the workload model around the notion of update bursts.

Using this terminology we can say that each instability creator is either generating a single update burst, in case of a prefix, or a set of update bursts, in the case of an AS. For example we can express BGP protocol divergence [5, 22, 44] for a prefix as a single update burst that lasts for the duration of the workload (or until it is fixed) and consists of a large number of updates.

Prefix forest

Routing updates are applied to and may change the existing routing table. Each routing/forwarding table consists of a set of prefixes. In abstract terms prefixes can be viewed as nodes in a forest. Each node covers a certain address space and a prefix is a descendant of another prefix if it covers a subset of the address space. A prefix S (son) is a child of another prefix F (father) if no other prefix exists that covers a larger address than S but a smaller address space than F. A possible root node for all prefixes corresponds to the full address space (the default route: 0.0.0.0/0). If the default route is present the table corresponds to a tree, otherwise to a forest. Figure 2 shows an example of a set of prefixes and their forest.

Once we view prefixes as forest we can use graph terminology to describe its properties. The *fanout* of a prefix is the fanout of its node in the forest, which is the number of its children. Intuitively, the fanout of a prefix specifies how many prefixes are more specific than a given prefix. The *depth* of a prefix is the depth of the corresponding node in the forest, which is the number of ancestors on the path from the node to the root of its subtree (including the root node). Intuitively the depth of a prefix specifies how often this prefix is a more specific prefix of another prefix. The *distance* of two prefixes, whose nodes in the forest are son and father, is the absolute difference of their prefix mask lengths. Intuitively the distance specifies how much more specific a prefix is.

The prefix structure in BGP tables reflect address allocation, aggregation and traffic engineering policies in the Internet. These policies have led to dependencies between the prefixes which is reflected in the structure of the prefix forest. Prefixes in the same subtree of the prefix forest are more likely to be correlated in terms of attribute values than two random prefixes. Actually, this is one of the reasons why the prefix structure influences the memory needed for storing the BGP tables on routers. A better understanding of the prefix structure may lead to better packet lookup and classification algorithms [45].

AS path properties

The AS path properties that are important for capturing the correlations between routing instabilities as observed by a peer include the properties of the ASes themselves and the peering policies reflected in the path. The AS properties include the distributions of the number of originated prefixes and transiting prefixes. These have been shown to be consistent with heavy-tailed distributions [36, 46, 47]. A small number of tier-1 ISPs provide transit for a huge number of prefixes, while a huge number of customers provide transit for none, and a sizeable number of tier-3 ISPs provide transit for some number of customers.

In general the peering policies between all ASes determine the AS level topology of the Internet. From the view point of a single router this general graph is restricted to an, at least ideally, directed acyclic graph (DAG) of the BGP announcements in its routing table (ignoring replicated AS

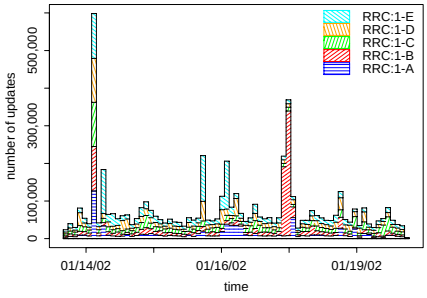

Fig. 4: # of updates for four peers.

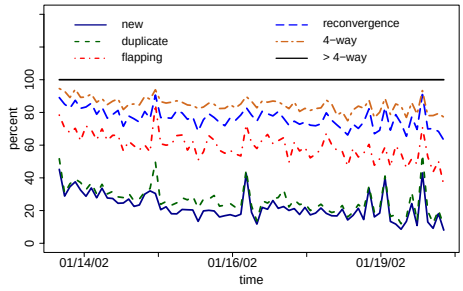

Fig. 5: Relative # of updates: x-way change.

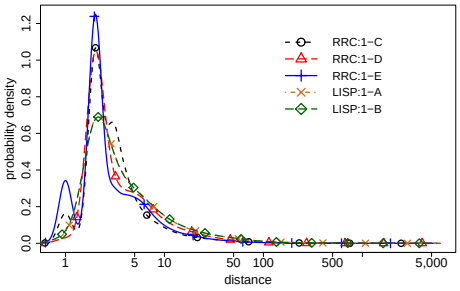

Fig. 6: # updates between updates with same set of attrib..

entries[4]). From the view point of a workload model we do not want to include the whole set of policies and the full topology. Rather we want to capture the important dependencies. Here we point out an interplay between the number of prefixes that an AS is transiting and its position in the DAG. The default policy for a router peering with a tier-1 ISP and a small local ISP is that a large number of its best routes will be via the tier-1 ISP and only a small number via the small ISP. If the router is only peering with the small ISP or it prefers the small ISP then a huge number of its best routes will be via the small ISP. Instead of incorporating all BGP policies and all BGP peering agreements into our workload model we propose to "just" incorporate the effects as observed by a BGP peer. This implies that we are not interested in which AS is peering with which other AS. Rather we are interested in the distance of the ASes from the peer and the number of originating and transiting routes. The latter matters since routes are coalesced at intermediate routers of the DAG and aggregated routes may be added. The richness of the connectivity does not get lost by considering just the distances either. The fanout of the DAG is captured in the number of ASes at a certain distance. In addition the number of nodes at each distance limits the number of alternative paths that may be explored by an update for a prefix originated by a distant router during route convergence. Traffic engineering and routing policies are reflected in the announced routes, AS replications on the AS path, and other attribute values.

In summary, we do not consider the full AS topology. Rather we propose to use the following ingredients in our BGP workload model: position of ASes in the BGP DAG, distribution of transiting/originated prefixes and distribution of AS path replication.

Attribute changes

Whenever we create an update for some prefix we need to decide if this is a new prefix or which attribute and in what fashion the attribute is to be changed. Some attributes are almost fixed, e.g., originator, others reflect the policies of the peer, e.g., communities and multi exit discriminator (MED), others reflect the convergence process, e.g., AS path and community. Which attribute changes we want to consider depends on the test-lab scenario that we have in mind. Certain test-lab scenarios might imply certain attribute values. For example the next-hop attribute is fixed for external BGP sessions. Other attributes might be uninteresting for a specific test-lab experiment, e.g., communities and MED, if the router under test ignores these values. On the other hand one might want to study in a test-lab what would happen if the peer starts to export community and MED attributes.

In general we propose to distinguish announcements from withdraws, and for consecutive announcements we either note the kind of change or the value change. The distinction is necessary in order not to disturb the other abstractions. For example for changes to the AS path it may not matter which AS is added to/deleted from the path. What matters is that the length is increased/decreased/constant. On the other hand a community encodes a certain policy of the AS and one might want to keep the meaning of the specific value.

It is important to consider attribute changes not just between two consecutive updates but over some number of updates. This is especially true within update bursts. Overall we note that the kind and the number of attribute changes capture some aspects of the dynamics of the convergence process and therefore the latency of route convergence. Another point that we need to deal with when considering multiple updates for the same prefix is timing. How much time separates two consecutive updates and how much time passes until two updates with the same attributes are observed. The latter corresponds to the time until the original route has been restored. To understand how many updates are involved before a route gets restored we propose to use the notion of an n-way change. An n-way change refers to a set of $n+1$ consecutive updates, where the last and the first updates are the first updates with the same attributes values.

Overall attribute changes and update burst are closely related to each other. Update burst capture more of the temporal characteristics while attribute changes capture the structural relationships between updates.

4. CHARACTERIZATION

Having discussed the key ingredient of our workload model we need to understand the probability distributions needed for the workload model. Contrary to other areas, including Web [48], telnet [28, 29], etc., the statistical properties of many of our workload ingredients have not been characterized before. In this paper we characterize both dynamic BGP traffic (Section 4.2), e.g., attribute changes/update burst/session resets, as well as static BGP tables (Section 4.3, 4.4), e.g., prefix structure, AS path. While the ingredients of the workload model are derived top-down the characterization has to proceed bottom-up. This implies that our anal-

[4]Replication is a traffic engineering instrument used to make a path less desirable.

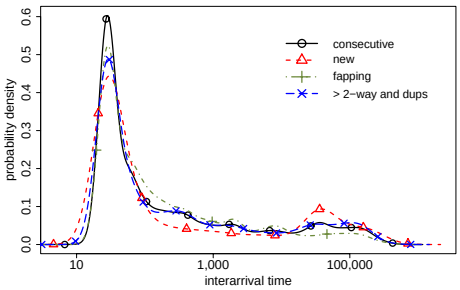

Fig. 7: Interarrival time: x-way change.

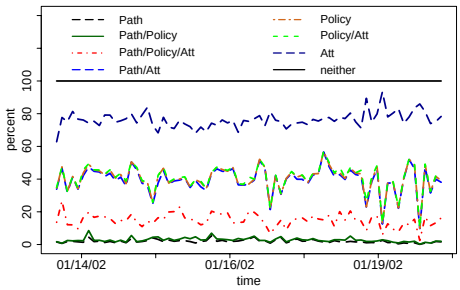

Fig. 8: Relative # of updates: attribute change.

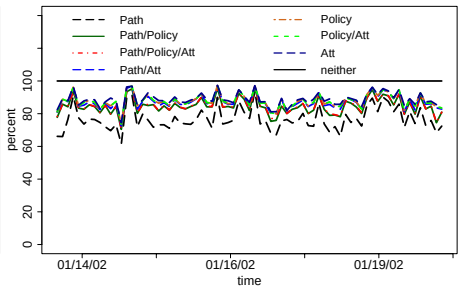

Fig. 9: Relative # of updates: attribute change.

ysis starts with the dynamics of the updates, then moves to sets of related updates, followed by the prefix forest, and ends with the AS path.

4.1 Data sets

Our characterization work is based on raw external BGP routing table dumps and update traces that we obtain from Ripe [49], SaarGate, a local ISP [50], Routeviews [51] and Merit [52]. Throughout this section we only present results in an exemplary fashion for the following raw data sets. **BGP update traces**: RRC:1 refers to the trace from RIPE's Remote Route Collector (RRC00) [49] (from 01/14/02, 1am to 01/20/02, 1:10am). It consists of 577 BGP update files with $8,442,000$ updates from 13 different peering session, including Tier 1 ISPs and major European ISPs. LISP:1 was gathered via two peering sessions with a local ISP, SaarGate (12/23/01, 10:05pm to 12/28/01, 1:05am). It contains roughly $959,000$ updates in 794 update files from two peering sessions. A lower bound on the number of missing updates is estimated by applying the updates to the starting BGP routing table and computing the differences between the resulting and the final routing table. We found 44 (4446) missing updates in LISP:1 (RRC:1). **BGP routing table dumps**: RRC:2 refers to all BGP table dumps, every 8 hours, from RIPE's RRC00 (from 12/31/01, 11:36pm to 01/08/02, 3:50pm). LISP:2 refers to all BGP table dumps from the local ISP during the same time period. Depending on the peer a BGP routing table consists of $90,300$ up to $109,200$ entries. To eliminate artifacts due to routing table fluctuations we calculate all table statistics for each table and then consider the mean over all table dumps.

4.2 BGP updates

The purpose of this section is to characterize BGP dynamics. Accordingly we start with the relationship between two updates for the same prefix then move onward to update bursts and finally propose a method for identifying session resets. In summary, we characterize BGP updates with respect to the metrics shown in Table 1.

BGP updates: Figure 4 shows the **number of updates for each two-hour period** for four peers for the week long data set RRC:1: First, some events, such as a failure of the collection machine, e.g., the first large spike, relative to time, can create updates that effect all peers. Other events can create a large number of updates that influence some subset of the peers. To eliminate artifacts due to errors in the data collection process we eliminated all updates caused by resets of sessions with the collector. Second, peers experience different numbers of updates during the same time period. This indicates that local peering policies and peer location influences the frequency as well as the kinds of observed updates. Third, all peers seem to show an overall similar behavior in terms of update rates, except during major peaks. Fourth, we observe few withdraws. Fifth, most announcements change some BGP attribute, especially after session resets with the collector have been removed. This confirms that the number of pathological instabilities (Labovitz et al. [14]) has been reduced substantially.

While consecutive duplications are not all that common, duplicate updates (*same set of attributes*) are rather frequent, due to flapping, etc. A flapping prefix using some route may cause the announcement of a new route, followed by the old route, followed by the new route, ... To characterize the process of convergence we want to know how many updates appear before the same update is repeated. To capture this we propose to use the concept of new, duplicate, flapping, re-convergence, and n-way change. Based on the count of intermediary updates we call an update a *new change*, if this is the first time an update for a prefix with this set of attributes is seen. It is a *duplicate*, if it is the same update as the previous update for this prefix. An update is a *flapping change*, if there is one update in between two updates with the same set of attributes. It is a *re-convergent change*, if there are two updates in between and an *n-way change*, if there are $n-1$ updates in between. Figure 5 plots the **stacked relative distribution of updates over time** according to this classification for data set RRC:1 and peer C, starting from new at the bottom to $>$ 4-way at the top. We first note that even during the later parts of the week new attribute sets are introduced. Even though we observe only 0.4% new prefixes, 20.8% new attribute sets are introduced. While we observe a trend towards a smaller number of new changes, it is apparent that during certain time periods spikes of new attribute sets are observed. During most time periods we observe few duplicates, and a large fraction of flapping prefixes (29.7% of the total). The fraction of updates that observe more than 2 updates in between is rather substantial (33.1% of the total).

Figure 6 shows the **density of the logarithm**[5] **of the number of updates in between reoccurrence of the same updates** for peers C, D, E and A,B of the RRC:1 and the LISP:1 data set. The fact that the maximum number of updates in between is larger than $1,000$ and that we need to plot this on a logarithmic scale is in itself rather amazing. Overall this indicates that at times the time for routing

[5]Coupled with a logarithmic scale on the x-axis, plotting the density of the logarithm of the data facilitates direct comparisons between different parts of the graphs based on the area under the curve.

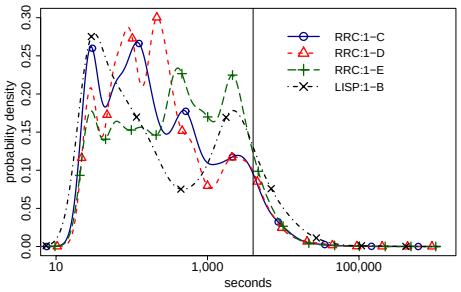

Fig. 10: Duration of bursts.

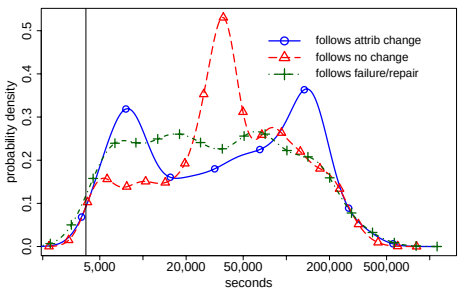

Fig. 11: Burst interarrival time.

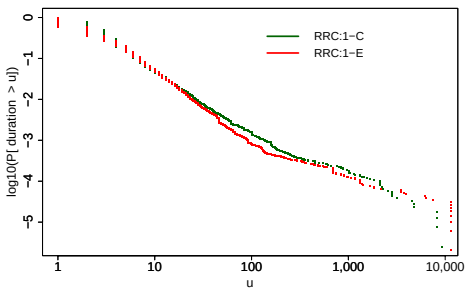

Fig. 12: updates per bursts.

convergence is quite long. This is consistent with the experiments from Labovitz et al. [17]. This is also confirmed by the distribution of the interarrival times of routing updates.

Figure 7 shows the **density distribution of the logarithm of the time between an update and the previous update** as well as the interarrival time of updates with the same set of attributes for peer C of data set RRC:1. From the density for consecutive updates we can see that the interarrival times of most updates is rather small, roughly 30 seconds which corresponds to a typical setting of the MinRoute Advertisement Interval timer [31]. Indeed 53.1% of all interarrival times are smaller than 60 seconds but bigger than 20 seconds, indicating that the MinAS Origination Interval timer, with a typical value of 15 seconds, is not influencing the spacing of updates. The probability curve flattens as the interarrival time reaches values of around 15 minutes, indicating that the time for most routing changes to take effect (except for route flap damping) is less than 15 minutes. Effects of route flap damping can be seen at $3-10$ minutes (first stage) and 30min to 1 hour later stages. Interestingly more than 11.8% of all interarrival times are larger than 12 hours, an indication for changes that remain stable.

The interarrival times between new updates and their previous update is quite different from the distribution for consecutive updates. Many of the new updates are stable updates and fewer have interarrival times less than 60 seconds and they are less likely to suffer from damping. The difference between the interarrival times of flapping prefixes and consecutive prefixes is that interarrival times of flaps are much more likely to be in the $1-10, 30-60$ minute range, indicating that they are more likely to be subject to route flap damping.

The next question is what is changed by an update. We distinguish between changes to the AS path and changes to other attributes, such as community, the later are denoted as "Attr". Changes to the AS path that are only due to policy considerations, e.g., duplication of ASes on the path, are called "Policy" changes; other changes to the AS path are "Path" changes. Figures 8, 9 show the **stacked relative distribution of updates over time** for peer C, peer D of data set RRC:1. We observe that most updates cause changes to the AS path. For some peers, e.g., peer D, most updates involve only the AS path. For other peers, e.g., peer C, combinations of path changes and attribute changes explain most updates. This depends on the policy of the peer. If the peer announces communities or other attributes, that can be used to influence routing policy decisions, as in the case of peer C, attribute changes are more prevalent, as if the peer does not announce such attributes. Changes to the AS path can either result in longer (35.3%), equal (31.2%), or shorter (33.5%) path lengths (for peer C). The fact that most updates contain larger or equal path length indicates that convergence after failures and routing changes are dominating the update process.

BGP update burst: After characterizing individual updates within their context we now move to understanding the correlations between updates. Accordingly we group updates for each prefix into update bursts in the same way as one groups packets into flows. If a peer sends two updates for the same prefix within a short time window, defined via a *timeout*, they are considered to be part of the same update burst. Based on the results by Varghese et al. [35], we use a timeout value of a bit larger than one hour (4000s). Our motivation is that each update burst should summarize all updates caused by one or multiple instability events and therefore capture the BGP convergence process.

We are interested in understanding the characteristics of update bursts such as arrival process, duration, number of updates. Correspondingly Figures 10, 11 show the **density of the logarithm of the duration and interarrival times of update bursts** for peers C/D/E (B) of data sets RRC:1 (LISP:1). While a specific timeout value changes the curves, we found that the general characteristics do not change. Route flap damping explains the various spikes of the different peers at 10, 15, 30 minutes. Since each peer is free to use its own parameters for their routers the values can differ quite a bit and are biased by the last peer along the path. Yet larger damping values will prevail and accordingly its not surprising that all peers have a spike at 30 minutes. Surprisingly the median duration of update bursts is rather small with 113 (87) seconds for RRC:1-C (LISP:1-B) and the 90% and 95% quantiles are less than 13.5 (15) and 19.5 (23.2) minutes. But the maximum durations are surprisingly large – they span the full trace duration. The only possible explanation is that some prefixes are constantly experiencing updates.

While the distribution of the durations of the individual update bursts is consistent with heavy-tailed distributions the distribution of the interarrival times is not. Figure 11 plots the **probability density of the logarithm of the interarrival time distribution** for peer C of RRC:1. To find such interarrival times we need at least two update bursts per prefix. Observing multiple update bursts indicates that multiple instability events occurred that each lead to a "stable" route. To understand how these instability events are related we distinguish three cases: **no change** – both bursts converged to an update with the same attributes (65.4%), **failure/repair** – one burst ended with a withdraw and the other with an announcement (12.4%), **attribute change** – the bursts end with announcements with different

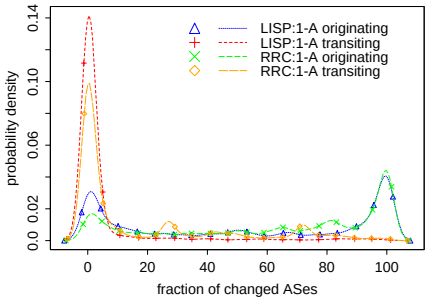

Fig. 13: % prefixes with updates

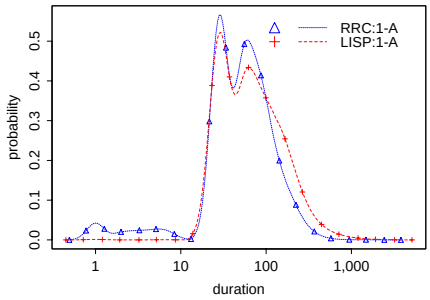

Fig. 14: Duration of session resets.

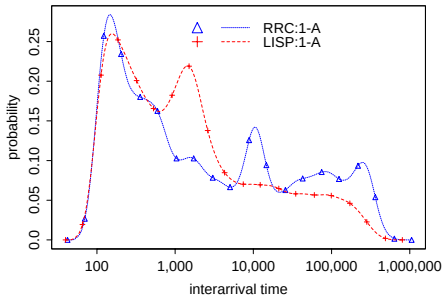

Fig. 15: Session reset interarr. time.

attributes (22.2%). The fact that a lot of consecutive update bursts end with the same update indicates that most prefixes, even if they experience an instability event, converge to a main route. Note that this indicates that most instabilities last only for a short time period and are contained within an update burst, which validates our methodology. Some failures cannot be repaired automatically and therefore involve two update bursts (captured by failure/repair). Here we cannot expect certain prevalent time periods which is supported by the interarrival time distribution. For update bursts that result in different attribute values we notice two spikes. Both spikes contain a large percentage of updates with an AS path of the same length. But the relative fraction of updates with longer paths is bigger in the first spike than in the second one. This is not surprising either since after a failure one can expect an alternative route announcement. Interesting features are the peaks in the distributions at time periods, $10-14$ hours and $1-2$ days, known from human behavior periods. The interarrival time distribution is not consistent with a heavy-tailed distribution. In contrast, the distribution of the **number of updates within each update burst** is consistent with heavy-tailed distributions. (Figure 12 plots the complementary cumulative distribution (ccdf) of the number of updates in a update burst.) Some bursts contain more than $10,000$ updates and some bursts have more than $1,000$ different updates. Furthermore the duration of a burst is strongly correlated to the number of updates (correlation > 0.92).

Possible session resets: In Section 3 we argued that an instability creator affects either a single prefix or many prefixes from the same AS. Indeed we argued that in the latter case the updates are or look like session resets. This also is confirmed by the presumption that a significant part of the routing table updates are results of *BGP session resets* [21]. A reset causes the two involved peers to update their BGP routing table and select a new best path for each prefix. If this results in a new "best path"/"no path" for a prefix, an announcement/withdraw has to be sent to all other peers.

A session reset on an access link connecting ISP A and B implies that all prefixes from A are now unreachable or reachable via some other AS. A session reset of the only peering link between two ISPs C and D implies updates for all prefixes that uses the ASes C and D on their AS path. But most ISPs peer at more than one location, see Figure 3. This implies that we will only see updates for prefixes whose best path included the link with the failed BGP session but not for all prefixes. Assuming that prefixes are evenly distributed across peering links, this explanation predicts updates at fractions such as 1/4, 1/3, 2/3 of the prefixes originating and transiting the ASes. Presumably the richer the peering connectivity of an AS is, the smaller is its reliance on any single BGP peering session and therefore the number of best path changes. Therefore we presume that the fraction of prefixes with routing updates decreases as the distance between the AS and the measurement point increases since the connectivity increases with the distance. Depending on the policy (export of IGP metrics via communities or MEDs) of the ISP a session reset of an IBGP session may not change the best BGP path or may change a substantial fraction of the best paths. This either causes only a small number of updates or a sizeable number of updates. For example in the example shown in Figure 3 an IBGP session reset will cause no updates as long as no advanced BGP features are used. In summary, we presume that most session resets result in routing updates for a significant fraction of the prefixes of the AS/ASes involved in the reseted BGP session.

To understand the magnitude of these fractions we want to compute, for each peer and for each AS, the fraction of updated prefixes associated with this AS within some small, appropriately chosen, time period. We approximate this value by computing, for each update and for each AS on the AS path, which fraction of the prefixes transiting/originated by that AS has changed during a window of plus/minus 3 minutes relative to the update timestamp. Averaged over 5 minute periods this approximates the fraction of changed originating/transiting prefixes for each AS. Figure 13 plots the **density of these fractions**, in percentages, for peer C (A) of data sets RRC:1 (LISP:1). For originating ASes the distribution is rather bimodal, either close to 80% or smaller than 20%. Therefore we only call events, where at least 80% of the prefixes originating at an AS saw updates, a possible session reset, in accordance with the above reasoning. Due to the high degree of peering at most transit ASes, each transit AS session reset is bound to only affect a much smaller fraction of the prefixes. This is reflected in the distribution which shows small peaks around 1/4, 1/3, 2/3. Given the shape of the distribution we only identify an update as belonging to a possible session reset, if at least 20% of the prefixes associated with a transit AS experienced updates.

After identifying which updates are part of a possible session reset, we group the updates into session resets in the same way that we grouped updates into update bursts. We explicitly choose a much smaller timeout value, 90 seconds, for session resets than for update bursts, since session resets are events localized in time. Experiments have shown that the results are not sensitive to this specific value. Figures 14 and 15 display the **density plots of the logarithm of**

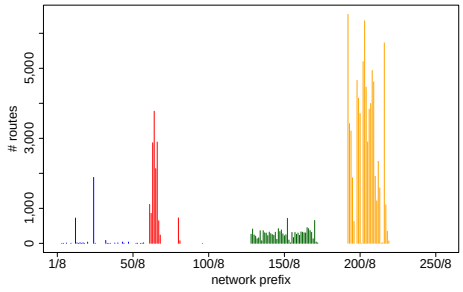

Fig. 16: # of routes per IP address range.

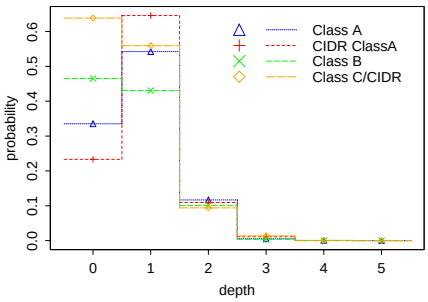

Fig. 17: Prefix depth.

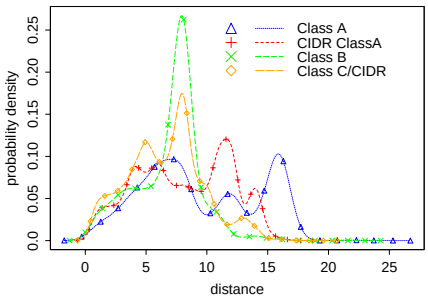

Fig. 18: Prefix distances.

the duration and interarrival times of session resets for the peer A (A) of data sets RRC:1 (LISP:1). Surprisingly some possible session resets last for a long time (> 30 minutes). Note that due to the 90 second timeout this implies a continuous sequence of updates. Such updates are likely caused by persistently flapping interfaces not subject to route flap damping. The fact that quite a large fraction (18.4%) of the session resets lasts longer than 90 seconds (second spike), indicates that routes are propagated along different paths. Some follow one AS path, others follow another longer or shorter path. Along the path they are subject to the Min-Route Advertisement Interval. This explains session reset durations of 90 seconds. The interarrival time distributions show the effect of synchronization due to route flap damping (different route flap damping parameters are in use by different ASes). After a session reset some routes may be subject to damping. After some time they all will be eligible for new updates again. To us, the observer, these later updates look like another session reset. Therefore it should be noted that we only identify candidate session resets. We are neither able to capture all session resets, nor are all captured ones actual session resets. But manual validation shows that the methodology is promising. Again the interarrival time distribution is neither consistent with an exponential distribution (too many dependencies), nor with a heavy-tailed one. On the other hand the number of updates within a possible session reset is, just as the number of (unique) updates within a update burst, consistent with a heavy-tailed distribution, although maybe not always a Pareto distribution.

4.3 BGP prefix forest

The structure of the prefixes in the BGP tables reflects the history of address allocation policies in the Internet. This policy has led to dependencies between the prefixes which is reflected in the structure of the prefix forest. We therefore analyze the BGP tables according to the properties of the prefix forest: fanout, depth and distances (for definitions of these terms see Section 3). To capture the history of classful address allocation we analyze the difference between the usage of the address space [53] of class A, B, C and CIDR blocks. Furthermore we consider the relationship between prefix length and distance in the prefix forest. In summary, we characterize routing tables with respect to the metrics shown in Table 2. The forest metrics are crucial if we want to study how routing instabilities influence forwarding performance (see [45, 54]). In contrast to the work by Huston [55] we are not analyzing the reasons of the growth rates of BGP routing tables, neither are we analyzing the long-term churn of BGP routing tables [56].

To highlight the dependency of the routing table on the history of address allocation policies, Figure 16 plots **how many prefixes exist with the same first octet of the IP address**. Intuitively this represents the usage of address space within each /8 prefix[6]. Looking back to the days of classful routing [53] class A networks should only announce a single prefix, class B networks should announce a maximum of 254 prefixes, and class C networks may announce up to $64K$ prefixes for each first byte of the IP address. This is clearly not the case today. For example within the former class B address range we usually observe between 100 to 254 prefixes. Still 23 of 45 groups within the same /8 prefixes announced slightly larger number of prefixes. The variability in terms of announced prefixes within each /8 is much larger in the former class C address range and extreme in the class A/A-CIDR range. While only 41 of the 126 possible class A networks did announce any prefixes, there are some that announce a lot of more specific prefixes, e.g., 12/8 (AT&T), announces over 650 more specific prefixes, or 24/8 (designated for data-over-cable networks), announces up to $1,970$ prefixes. Some of these more specific prefixes are used to implement the CIDR allocation strategies, others appear to be used for routing policies such as multi-homing, traffic splitting/sharing, load balancing, etc. Due to the large differences we study the forest metrics separately for each address range: class A, B, C and A-CIDR.

Figures 17, 18 confirm this decision. The plots show the **density of the depth and the distances of the prefixes** separated according to address ranges. The **depth** of the prefixes (see Figure 17) reflects how many holes are punched into the address space, i.e., how often one address block is more specific than another one. Surprisingly, we have observed that more than 84 prefixes have depth $4-5$, a rather large number. Just because a prefix is at depth 4 or 5 does not imply that it is a point-to-point link (/30) or a host route (/32). Ironically no /30 or /32 can be found at depth 4 or 5 (indeed over 80% are nested exactly 1 time). Rather we observe that for prefixes with large depth the distances between the specific routes are rather small, resulting in progressions of, e.g., /17, /19, /20, /21, /22. This suggests that this technique is used for traffic engineering and multi-homing. Punching holes is most extreme in the class C address block.

The distribution of **prefix lengths** is as expected. Less

[6]Note, that not all prefixes with a mask of /8 need to be present in the routing table.

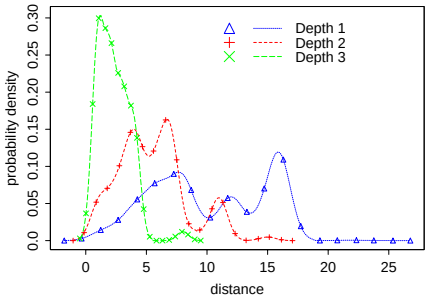

Fig. 19: Prefix distances (class A).

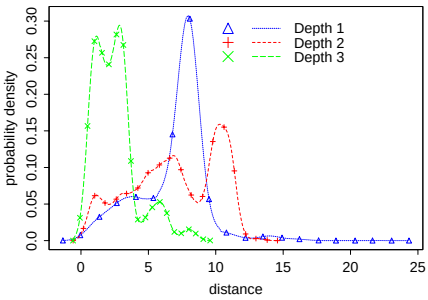

Fig. 20: Prefix distances (class B).

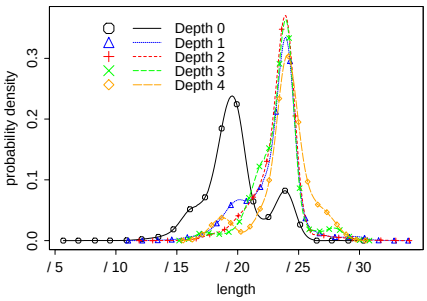

Fig. 21: Prefix length (A-CIDR).

than 3% of the prefixes are larger than 24 and more than 54% of the prefixes have length 24. Prefixes from class B are mostly /16s or /24s with some prefix length in between. Class B is the only address block where another prefix length, in this case /16, is more prevalent than /24. For class C we observe mostly /24s with some number of /19s and /20s and a smaller number of prefixes between 20 and 24. Class A and A-CIDR (e.g., see Figure 21) prefixes have relatively speaking the smallest number of /24s and the largest number of /19s, /20s and /21s−23s. For example the peak at /19 reflects a common filter policy, some ISPs filter prefixes with more specific prefix mask than /19. This forces others to allocate at least a /19 address range to be visible in the whole Internet.

The characteristics of the prefix length and the depth of a prefix significantly influences the distribution of the distances between prefixes. Figure 18 shows the **density of the distributions of the distances for the various address blocks**. As expected the distance 8 dominates the distribution with more than 20% of the total. While overall distances of less than 8 are very frequent, $1, 2, 3$ with roughly 5% and $4, 5, 6, 7$ with roughly 8% each, are the most prominent distances for class A (class A-CIDR) prefixes are 16 (12 and 14) reflecting the different policies.

With regards to the **fanout** we have observed (not shown) that A-CIDR has the largest fanout followed by class C and A. Class B prefixes in general have smaller fanout than other prefixes. Overall the tails of the fanout distributions are consistent with heavy-tailed distributions such as the Pareto distribution. This indicates that the density distribution might be biased by a few providers using a large fanout.

Naturally, each of the forest metrics does not just depend on the address block of the prefix, but also on the depth of the prefix in the tree. For example, Figures 19 and 20, show the **density of the distance of the class A and class B prefixes for each prefix depth**. While the distributions for the prefixes at depth 3 are similar, the distribution of distances for depth 1 and 2 are quite distinct. For both classes depth 1 is dominated by /16 or rather /8, which brings us to /24 prefixes, the most specific prefixes that are allowed by most ISPs. The differences are most likely results of folks with class A prefixes taking more advantage of the available address space and using intermediary aggregation levels. This is reflected by Figure 17 which shows that prefixes in class A are more likely to be at higher depth. At the beginning the results for class B and depth 2 seem counterintuitive. But the peek at distance 11 is the result of classless routing even within the class B address block. Otherwise the distribution at depth 2 reflects the flexibility of sub-netting within class A/B networks.

While the dependencies between the metrics, especially on the depth of a prefix can be significant, e.g., for the fanout, each metric behaves differently. For example Figure 21 shows the **density of the prefix length for class A-CIDR for each depth**. Here we observe the artifacts of the class A network at depth 0. But the prefix length distributions at other depths are very similar to each other. Actually, it is remarkable how little difference there is for depth $1 - 4$ given the differences in the prefix distance distributions.

4.4 AS path

Instead of trying to understand the AS-graph level Internet topology, we need to characterize the AS path with respect to the ingredients for our BGP workload model. Accordingly we consider the AS-graph from the view point of the peer, a BGP DAG, and capture the position of each AS by the distribution of the distances of ASes from the peer. In addition we need to consider the number of originated and transiting prefixes[7]. Furthermore we explore some of the basic characteristics of the AS path, e.g., the length of the AS path, the number of unique ASes on the AS path, the positions of replicated AS on the AS path, and the number of replicated AS entries. This results in a characterization of the AS path according to the metrics shown in Table 3.

Since session resets are a prevalent reason for routing updates we need to understand how many prefixes are transiting or are originated by an AS. Figure 22 plots[8] the **density of the logarithm of the number of prefixes originated/transiting an AS** for the data sets RRC:2 and LISP:2. It is not surprising that the distributions from different peers are almost equal, because default-free routing tables contain most of the prefixes of the Internet. A much larger number of ASes are providing connectivity to only a very small number of prefixes. Well connected peers, closer to the center of the topology, provide connectivity for a lot of other prefixes and ASes. This is reflected in the tails of the distributions which are consistent with heavy-tailed distributions (not shown). Still, for a workload ingredient, the shape of the body of the distribution is at least as important as the tail of the distribution. Note that the distribution of the transiting ASes reflects the coalescing of AS paths at intermediary routers.

As argued above we do not care about the exact sequence

[7]A prefix is transiting an AS if an entry/update for this prefix uses this AS on its AS path.

[8]The dip in the originating curve is due to discretization effects.

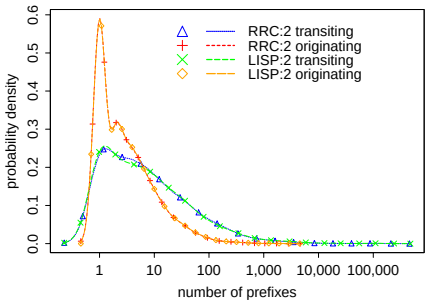

Fig. 22: Distribution of prefixes per AS.

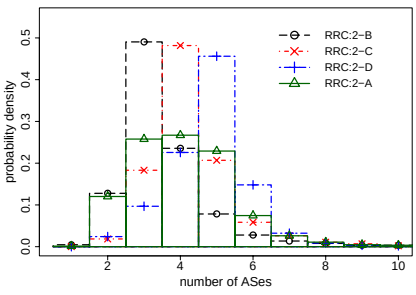

Fig. 23: Distance of AS from peering point.

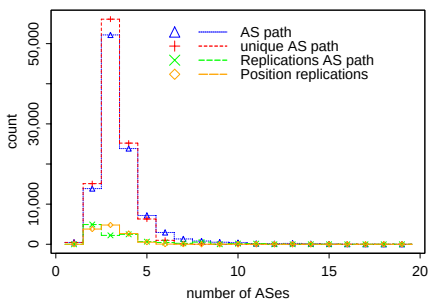

Fig. 24: Length of AS path.

of ASes on the AS path, only about locating an AS at approximately the right distance from the peer. (Note the distance influences the routing update latencies via intermediary ASes, the richness of the connectivity, etc.) Therefore we, for every AS, calculate the mean distance of this AS to the BGP peer in numbers of unique AS hops on the AS path of the prefix[9]. While this distance does not have to be non-ambiguous, we observed, that it is non-ambiguous for 85% of the ASes. The standard deviation of the ones that are ambiguous is usually less than 0.5 and the standard deviation relative to all ASes is less than 0.01, a rather small value. Figure 23 plots the **relative frequency of the distances for all ASes** for four of the data sets RRC:2. For some peers the curves appear to be shifted by one or two. For others the distributions are more bell-shaped. This indicates that connectivity characteristics depend very much on the location of the peering point in the Internet hierarchy. This is due to the different distances to the "core of the Internet", e.g., tier-1 providers. If a tier-1 provider is reachable within a short distance then most of the ASes will be reachable within a slightly larger distance due to the huge connectivity of tier-1s [57, 38].

In terms of general characteristics we find that while most paths are short (93.5% are less than 6 AS hops) some are sizeable (0.75% are greater than 10 AS hops). Figures 24 shows the **histogram of the number of ASes on the AS path**. Eliminating replicated ASes from the AS path reduces the average AS path length from 3.5 to 3.2 affecting 10.5% of the AS paths. The median length of the replicas is 3. There are many short replications of 2 or 3 replicas but also some rather long ones with 8, 10, or 11 replications. The position of the replications on the AS path is rather early (not shown). Almost all duplications appear between position 2 and 5, indicating that this instrument is mainly used in the center of the Internet. On the other hand most replications appear closer to the originator of the prefix indicating that the instrument is applied close to the edge of the network.

To understand the correlation between the location of a prefix in the prefix forest and its AS path we consider the similarities between the AS paths of parent and sons. We find that 26.5% of all nested prefixes have the same AS paths as their fathers. Furthermore 20.6% of the AS paths of nested prefixes just contain additional ASes. For example this happens if the AS number of a multihomed AS is added to the AS path. But 52.9% of all nested prefixes have a different AS path than their parent prefix (e.g., multihomed ASes with other upstream ISPs than the one responsible for their address space or an AS that has switched from one provider to another while keeping its address space). Depending on the routing table data structure such common parts can be used to optimize memory usage. This in turn explains some of the dependencies between router memory requirements and prefix forest characteristics.

[9]Note, that replicated ASes are eliminated from the AS path, before this metric is calculated.

4.5 Summary

Our results confirm that it is possible to characterize the proposed ingredients of a workload model via empirical probability distributions according to the metrics outlined in Tables 1, 2 and 3. We find that some of these cannot be easily captured by simple, one or two parameter, distributions. Therefore, at least for the moment, we propose to either rely on experimentally derived probability distributions or manually edited probability distributions to instantiate the workload. In addition, we have identified some important dependencies between the ingredients, e.g., the fanout distribution is dependent on the depth in the prefix tree. In such a case we propose to use combined probability distributions for both.

5. RTG: ROUTING TABLE GENERATOR

Having identified and characterized key ingredients of BGP traffic we can, in this Section, turn to a proposal for generating realistic BGP traffic and its prototype implementation, `RTG`. The main idea is the possibility of generating routing updates off-line, storing them in a file, and then feeding them to the system, e.g., a router under test, using a simple program that is capable of maintaining BGP sessions and sending BGP updates (see Figure 25). Accordingly the tool consists of two independent pieces: (a) a routing table generator (`RTG`) which generates routing tables and updates and (b) SBGP from the Merit toolkit MRTd [58] for feeding the updates to the system under test. Each entry generated by `RTG` is characterized by a timestamp, the originating peer, the prefix and its attributes. The timestamp specifies when the update is supposed to be issued by SBGP.

`RTG` itself consists of three parts. The first part is responsible for building a routing table and is parameterized in terms of size of routing table and characteristics (prefix length distribution, depth and fanout via configuration files). The second part of `RTG` associates each prefix of the routing table with a set of attributes. (This process is again driven by configuration files.) The output of the first two parts is used to instantiate the initial routing table. The

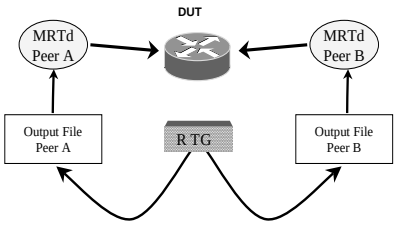

Fig. 25: Example: RTG scenario.

third part of RTG is responsible for generating the actual routing updates and is again driven by configuration files.

RTG prefix structure: The table generation piece is emulating the address allocation strategy in the Internet. The instantiation of the routing table proceeds top down, from the root of each tree in the forest to the leaves. The generation process starts by picking how many trees should be generated. The process of generating a tree starts by picking a prefix length l, followed by a network prefix of this length P/l. The requirement for the prefix is that it does not conflict with any previously chosen prefixes. This means that P/l is a more specific prefix only within the current subtree. The subtrees of P/l are generated recursively. The difference is that the new prefix lengths have to be larger than l and that the address range is limited by P/l. The prefix length and the fanout are chosen according to various empirical probability distributions for each level. At the first level the prefix is selected according to the distribution of routes within the former classful address ranges and uniformly at random for all other levels.

RTG attributes: Once a prefix has been chosen RTG selects attributes for it. The degree of choice depends on the attribute. For some there is none since it only depends on the physical test-bed setup, others are selected according to empirical distributions, e.g., community, while another attribute, the AS path, requires more care. The AS path attribute is essential for building the updates, since it approximates the underlying topology. We first generate an AS path pattern which specifies the length of the path, the number of duplicated ASes on the path, and the location of these duplications. This is constructed out of the probability distributions for the AS path length for the peer, the number of duplications and the position of duplicates. In a next step this AS path pattern is filled with ASes based on a probability distribution (transiting and originating ASes are constructed separately). If the prefix is nested within another the attribute values are either copied, modified, or newly constructed, based on a probability distribution specified in the configuration file.

RTG updates: The various kinds of routing updates are generated in two steps. First an event log specifies which events are created by the instability creators. Possible events for an AS instability creator are BGP session resets for some (random) AS, BGP session resets at some (random) AS path distance. Possible events for an prefix instability creator are update bursts for some (random) prefix, or single changes for some (random) prefix. Note that this file can be automatically generated based on the distribution of interarrival times of session resets and update bursts. Based upon the event log we construct a detailed list of updates. Each session reset implies that some fraction (chosen according to a probability distribution) of the prefixes (chosen uniformly) of a given or randomly chosen AS experience an update burst starting at the specified time. Each update burst is realized by selecting the number of involved updates and the update interarrival time according to the appropriate probability distribution or it may be taken from a trace. Next the updates are spaced within the available time frame. To instantiate a single update the attribute changes have to be selected. This is done according to the observed attribute changes: AS path changes, policy changes, or other attribute changes.

RTG advantages: RTGs approach differs from black-box approaches that just replay a trace, in that it captures the structure of BGP traffic. It is highly configurable, parameterizable and scalable and therefore allows insights into the reasons behind certain behaviors. Through its flexibility and parameterization RTG enables us to study the sensibilities of test systems in a repeatable and consistent manner while still providing the possibility of capturing the different characteristics from different vantage points in the network. On the other hand the ability of scaling a trace and/or adapting it to a different scenario is rather limited. Traces taken from different vantage points in the network can have significantly different characteristics and may or may not be appropriate for certain tests [23]. In addition traces are usually considered proprietary. But tools, such as RTG, can be used by people at different locations and companies, that normally could not share data to normalize their workloads [26]. In addition it allows us to take one characteristic from one trace and other characteristics from another trace.

Just consider two examples where RTG may help us: packet classification performance and interactions of routing updates and classification. Having a realistic routing table is crucial for the performance evaluation of the classification algorithm, e.g., to determine the memory requirements. Also the performance of some classification algorithms depends on the depth of a prefix in the tree. Each routing update has the potential to change the routing table and therefore interact with the forwarding performance. How many updates and what kind of updates are necessary to break a router depends on the structure of the updates and the size of the table [20]. RTG allows us to explore this relationship in a defensible fashion, instead of relying on the traces to contain the necessary event sequences.

6. VALIDATION

The goal of the workload generator should be to generate synthetic traffic that reflects the real workload as accurately as possible. Therefore its validation is crucial for interpreting the results derived from using the workloads. Ideally the validation of the workload model is not just a verification of the parameters of the workload but also a demonstration that the performance of a system subjected to the workload is similar to the performance of the system under a real workload. This is beyond the scope of this paper and will be subject of another paper which examines the behavior of a router subjected to trace-based and RTG generated workloads.

Our trace analysis has shown that BGP traffic has many dependencies that are in part captured by our workload model. To highlight some of RTGs features we examine, in an exemplary fashion, if the characteristics of the generated workload are consistent with the characteristics of the measured workload. For this we concentrate on derived measures rather than distributions that are already part of RTGs

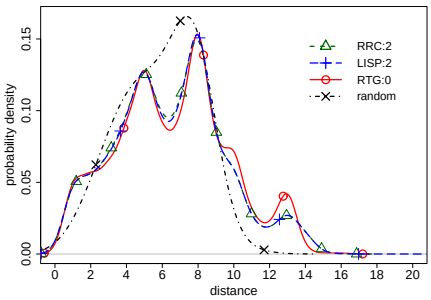

Fig. 26: Distance class C.

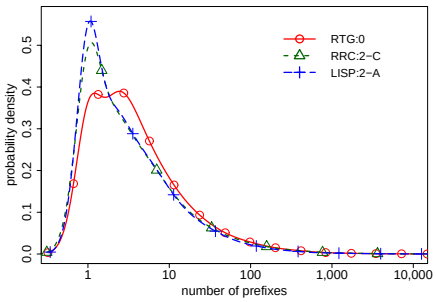

Fig. 27: Prefix dist. per AS.

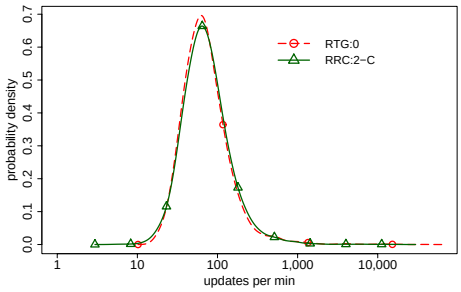

Fig. 28: Update rate per min..

configuration. We show one example for each of the three major components of RTG: BGP prefix forest, AS path, and BGP updates.

BGP prefix forest: Since distances are not used as parameterized distribution in the table construction process we present results about their distributions in order to verify the nesting, fanout and prefix length at the different levels. Figure 26 shows the **distribution of the prefix distances for the class C address block**, for a RTG generated table with roughly twice the number of prefixes (227,747) and for the two routing tables RRC:2 and LISP:2. One can observe good agreement even though the RTG routing table contains twice the number of prefixes as the original routing tables. The later underlines some of RTGs capabilities. In addition the plot also contains a routing table with IP addresses chosen randomly and prefix mask chosen according to the prefix length distribution. Here we observe a sizeable difference. Checks of the other metrics show good agreements at all depths.

AS path: Figure 27 plots the **density of the logarithm of the number of prefixes per AS** (this includes transiting and originating). We observe an apparent disagreement between the curves for the RTG generated table and the two datasets for small numbers of prefixes per AS. But one just has to recall that the RTG generated table contains twice the number of prefixes as the other two tables but the same number of ASes. If we double the number of ASes as well we again observe good agreement. The tails of the distributions (not shown) agree rather well.

BGP updates: The analysis of Section 4.2 has highlighted some of the dependencies between BGP updates. Therefore we cannot hope to find a single plot that validates this process. Rather we pick one kind of instability creators: prefix addition/deletion which each generates an update burst. We generate an event log with update bursts based upon the interarrival time distributions of update bursts. In a first step each update burst in the log file is replaced with an update burst of the same kind from the actual trace. Figure 28 shows the **resulting rate of updates** for both the actual update trace and the generated update trace (with twice as many prefixes). In a second step we generate the update bursts according to the burst characteristics (not shown). Note the good agreement of the rate plot.

7. SUMMARY

This paper motivated and presented our workload model for BGP traffic and its prototype realization, RTG. The workload model is based on the key notion of a BGP instability creator who creates correlated instability bursts via the AS path, and effects the BGP routing, characterized via the notion of a prefix forest. We show how to derive the distributional parameters of RTG from actual BGP tables and updates. Indeed we present an analysis of instability creators and instability events involved in the current Internet routing instabilities. In summary, we find that RTG is capable of generating realistic BGP traffic in the lab.

The development of our prototype tool is part of a larger research effort of bringing the variability of the Internet into test labs. The goal of the project is to study the impact of variability in a controlled environment. RTG adds an important component, routing, to the existing toolset of workload generators and traffic shaping tools. A test bed with all these components will not just enable us to answer the questions stated in the introduction but experiment with, evaluate, and judge most Internet components.

Acknowledgments

We are in debt to Jan Bankstahl of SaarGate and the RIPE RIS project for access to the BGP data. Numerous colleagues at Saarland University have provided valuable feedback on this work. We thank Jennifer Rexford, Andrew Moore, Mark Crovella and the anonymous reviewers for their detailed and insightful comments that greatly improved the paper. This research has been partly supported by Cisco Systems and the Defense Advanced Research Projects Agency (DARPA), under grant N66001-00-8065 from the U.S. Department of Defense. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of Cisco Systems nor the Department of Defense.

8. REFERENCES

[1] IXIA BGP Routing Protocol Emulation Software, 2002. http://www.ixiacom.com/.

[2] Chariot, 2002. http://www.netiq.com/products/chr/default.asp.

[3] TeraRouter Tester, 2002. http://www.netcomsystems.com/solutions/products/applications/pdf/TeraRouting/index.htm.

[4] A. Krämer, "RIG, A BGP Routing Instability Generator," 2002. Diploma Thesis, ETH Zürich. http://www.barman.ch/rig/.

[5] T. G. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. ACM SIGCOMM*, 1999.

[6] A. Shaikh, R. Dube, and A. Varma, "Avoiding instability during graceful shutdown of OSPF," in *Proc. IEEE INFOCOM*, 2002.

[7] T. Griffin, "What is the Sound of One Route Flapping?," 2002. IPAM talk.

[8] B. Halabi, *Internet Routing Architectures*. Cisco Press, 1997.

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

[10] C. Huitema, *Routing in the Internet*. Prentice Hall, 1999.

[11] B. Chinoy, "Dynamics of Internet routing information," in *Proc. ACM SIGCOMM*, pp. 45–52, 1993.

[12] R. Govindan and A. Reddy, "An analysis of Internet inter-domain topology and route stability," in *Proc. IEEE INFOCOM*, 1997.

[13] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," tech. rep., 96-631, USC/ISI, 1996.

[14] C. Labovitz, R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 515–558, 1998.

[15] C. Labovitz, R. Malan, and F. Jahanian, "Origins of Internet routing instability," in *Proc. IEEE INFOCOM*, 1999.

[16] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of Internet stability and wide-area network failures," in *Proc. International Symposium on Fault-Tolerant Computing*, 1999.

[17] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," in *Proc. ACM SIGCOMM*, 2000.

[18] T. G. Griffin and B. J. Premore, "An experimental analysis of BGP convergence time," in *Proc. International Conference on Network Protocols*, 2001.

[19] A. Shaikh, L. Kalampoukas, R. Dube, and A. Varma, "Routing stability in congested networks: Experimentation and analysis," in *Proc. ACM SIGCOMM*, 2000.

[20] D. Chang, R. Govindan, and J. Heidemann, "An Empirical Study of Router Response to Large BGP Routing Table Load," tech. rep., USC/ISI, 2001.

[21] S. Ramachandra, Y. Rekhter, R. Fernando, J. Scudder, and E. Chen, "Graceful restart mechanism for BGP," 2001. Internet Draft (draft-ietf-idr-restart-05.txt).

[22] D. McPerson, V. Gill, D. Walton, and A. Retana, "BGP persistent route oscillation condition," 2001. Internet Draft (draft-ietf-idr-route-oscillation-01.txt).

[23] H. Berkowitz, A. Retana, S. Hares, and P. Krishnaswamy, "Benchmarking methodology for basic BGP convergence," 2002. Internet Draft (draft-ietf-bmwg-bgpbas-01.txt).

[24] H. Berkowitz, A. Retana, S. Hares, P. Krishnaswamy, and M. Lepp, "Terminology for benchmarking external routing convergence measurements," 2002. Internet Draft (draft-ietf-bmwg-conterm-01.txt).

[25] NANOG: The North American Network Operators Group. http://www.nanog.org/.

[26] S. Hare, P. Krishnaswamy, M. Lepp, A. Retana, H. Berkowitz, and E. Davis, "BGP convergence measurement issues," 2001. IETF/bmwg talk.

[27] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *Proc. ACM SIGMETRICS*, pp. 151–160, 1998.

[28] P. Danzig, R. Caceres, D. Mitzel, and D. Estrin, "An empirical workload model for driving wide-area TCP/IP network simulations," *IEEE/ACM Trans. Networking*, vol. 3, no. 1, pp. 1–26, 1992.

[29] P. Danzig and S. Jamin, "tcplib: A Library of TCP Internetwork Traffic Characteristics," tech. rep., USC, 1991.

[30] C. Huitema, *Routing in the Internet*. Prentice Hall, 1995.

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

[32] C. Villamiyar, R. Chandra, and R. Govindan, "BGP route flap damping," 1998. RFC 2439.

[33] C. Panigl, J. Schmitz, P. Smith, and C. Vistoli, "RIPE Routing-WG Recommendation for Coordinated Route-flap Damping Parameters ," 2001. http://www.ripe.net/ripe/docs/ripe-229.html.

[34] M. Musuvathi, S. Venkatachary, R. Wattenhofer, C. Labovitz, and A. Ahuja, "BGP-CT: A First Step Towards Fast Internet Fail-Over," in *Microsoft Research Technical Report*, 2000.

[35] G. Varghese, R. Govindan, R. Katz, and Z. Mao, "Route flap damping exacerbates Internet routing convergence," in *Proc. ACM SIGCOMM*, 2002.

[36] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. ACM SIGCOMM*, 1999.

[37] K. Calvert, M. Doar, and E. W. Zegura, "Modeling Internet topology," in *IEEE Communication Magazine*, 1997.

[38] L. Gao, "On inferring autonomous system relationships in the Internet," in *Proc. IEEE Global Internet*, 2000.

[39] L. Gao and J. Rexford, "Stable Internet routing without global coordination," in *Proc. ACM SIGMETRICS*, 2001.

[40] B. Norton, "The art of peering: The peering playbook," 2002.

[41] "SSFNet, Scalable Simulation Framework." http://www.ssfnet.org/.

[42] O. Maennel and A. Feldmann, "BGPcharacter: Tool for processing and characterizing BGP data," February 2002. NANOG 24 talk.

[43] D. Wetherall, R. Mahajan, and T. Anderson, "Understanding BGP misconfigurations," in *Proc. ACM SIGCOMM*, 2002.

[44] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "Policy disputes in path vector protocols," in *Proc. International Conference on Network Protocols*, 1999.

[45] A. Feldmann and S. Muthukrishnan, "Tradeoffs for packet classification," in *Proc. IEEE INFOCOM*, 2000.

[46] W. Fang and L. Peterson, "Inter-AS traffic patterns and their implications," in *Proc. IEEE Global Internet*, 1999.

[47] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: Degreebased vs. structural," in *Proc. ACM SIGCOMM*, 2002.

[48] B. Krishnamurthy and J. Rexford, *Web Protocols and Practice*. Addison-Wesley, 2001.

[49] RIPE's Routing Information Service Raw Data Page. http://data.ris.ripe.net/.

[50] SaarGate. http://www.saargate.de/.

[51] University of Oregon RouteViews project. http://www.routeviews.org/.

[52] Merit. http://www.merit.edu/.

[53] V. Fuller, T. Li., J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," 1993. RFC 1519.

[54] Light Reading, "The Internet Core Router Test," 2001.

[55] G. Huston, "Analyzing the Internet BGP routing table," in *Internet Protocol Journal*, 2001.

[56] A. Broido, E. Nemeth, and K. Claffy, "Internet Expansion, Refinement and Churn," in *ETT*, 2002.

[57] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, 2002.

[58] C. Labovitz, "Multithreaded routing toolkit," in *Merit Technical Report to the National Science Foundation*, 1996.