Generating realistic routing tables in a test-lab

Olaf Maennel, Anja Feldmann

I. Introduction

Lots of new features are added to the Internet every day to improve quality of service. These features include RED, fair queuing, etc. Unfortunately testing and evaluation of the effectiveness of these mechanisms in test environments is often limited by the simplicity of the test setup. Yet, variability is necessary to locate equipment bugs and to evaluate the features benefits. As a result the real test is the deployment in the operational network - not a sound testing methodology.

As a first step towards addressing this problem, we are building a tool that can generate realistic routing tables of arbitrary size and updates of arbitrary frequency. To this end, we first define a set of metrics that can capture important characteristics of routing tables and routing updates. Next, we present insights of a characterization of measured routing tables and updates and identify some key insights. These results are then used to parameterize our tool. Finally, we describe the key ideas behind the tool and conclude with a brief discussion of future work.

II. Metrics

By realistic routing tables and updates we mean those that have the same characteristics as actual routing tables and updates with respect to the following metrics:

For the routing tables we consider the distributions of:

- routes within the IP address range
- prefix length
- depth and fanout[1] of nested routes

For updates:

- AS-Path
- number of updates per route
  (dependent on prefix length)
- interarrival time of updates and
  between updates on a per route basis
- IBGP vs. EBGP updates

III. Characterization of routing tables and updates

Our characterization is based on the analysis of publicly available routing tables from major NAPs (MAE-EAST, MAE-WEST and European Networks collected by Merit and Ripe). Since some of the characteristics are highly dependent on the connecting peer (and its allocated prefix range) we compute each metric for all entries as well as on a per peer basis.

Our analysis of the routing tables shows that there are strong dependencies between the IP address range, prefix length, nesting depth, and fanout. These dependencies arise in parts from the effects of CIDR, classfull routing, route aggregation, and route announcements of more specific routes. The latter are partly due to multihoming, and/or traffic engineering, and/or unfortunate design decissions, and/or typing mistakes. For example suballocation and subnetting of class A and B networks results in nesting depth of over 8 and fanout of up to 15000.

A significant part of the routing table updates are results of BGP session resets. Each reset causes the two participating peers to exchange their full routing tables and distribute the learned information to their other peers. Therefore the effects of routing table resets are visible even if the responsible peers are many AS-hops away. It appears that during some time periods a single AS is responsible for almost all of the updates.

IV. Tool and evaluation

The tool consists of two independent pieces: a) a routing table generator (RTG) which generates routing table updates and (b) the Merit tool, MRTd, that feeds these updates to the router under test. The latter part is ideal for execution in a test-lab.

RTG itself consists of three parts. The first part is responsible for instantiating a routing table and is parameterized in terms of number of routing table entries and routing table characteristics in terms of prefix length distribution, depth and fanout via configuration files. (These configuration files can be derived from actual routing tables or manually edited.) The second part of RTG associates each prefix of the routing table with a AS path and other attributes. This process is again driven by configuration files. The output of the first two parts is a list of routing updates that can be used to instantiate a routing table in a test router. The third part of RTG is responsible for the generation of actually routing updates driven by configuration files. The output is another list of routing updates. The main motivation for splitting the tool into these three parts are the strong correlations between routing updates and ASs on the AS-path as well as the strong correlations between prefixes in the BGP table and BGP attributes, specifically the AS-path.

The configuration files of the tool try to capture the essential characteristics of BGP. By changing some of the parameters it is possible to test scenarios in a test-lab. E.g. by changing the number of routing table entries and update characteristics it should be possible to answer questions regarding BGPs scalability.

V. Conclusion and future work

We evaluated the routing tables and updates generated by RTG according to the same metrics as discussed earlier and conclude that it is possible to generate realistic routing tables in a test-lab. We plan to continue our work by building a testbed consisting of a router, some computers which simulate BGP peers and other computers that generate traffic. We intend to use this testbed for answering questions regarding BGPs scalability and BGPs limitations, interactions of IBGP and EBGP, etc.

More information about the project can be found under:
`http:://www.net.uni-sb.de/~olafm/`

O. Maennel and A. Feldmann are with the Saarland University, Saarbrücken, Germany. E-mail: {olafm,anja}@net.uni-sb.de

[1] The nesting depth of a prefix specifies how often this prefix is a more specific prefix of another prefix. The fanout of a prefix specifies how many prefixes are more specific than a given prefix. More mathematically: Let prefix α denote $a.b.c.d/x$, prefix β_1 be $a_1.b_1.c_1.d_1/y_1$, ..., and prefix β_n be $a_n.b_n.c_n.d_n/y_n$ where $y_i > x$, β_i is a subprefix of α, and β_i is not a subprefix of β_j. x is the prefix length of α and y_i is the prefix length of β_i. In this case α has nesting depth 0, and β_i have nesting depth 1. Since α has n more specific sub-prefixes the fanout of α is n while the fanout of β_i is 0.