BGP Molecules: Understanding and Predicting Prefix Failures

Ravish Khosla, Sonia Fahmy, Y. Charlie Hu
Purdue University
Email: {rkhosla, fahmy, ychu}@purdue.edu

***Abstract*—The Border Gateway Protocol (BGP), the de-facto Internet interdomain routing protocol, disseminates information about Internet prefixes to Autonomous Systems (ASes). Prefixes are announced and withdrawn as routes and policies change, making them unreachable from portions of the Internet for certain time periods. This paper aims to predict routing failures of prefixes in the Internet. We investigate the similarity of prefixes in the Internet with respect to their propensity to fail, i.e., become unreachable. Given a prefix of interest, we define a "BGP molecule" – the prefixes in the Internet that are likely to fail together with this prefix. We show that the AS paths to prefixes, coupled with knowledge of the prefix geographical location, contribute to its failure tendency. The BGP molecules constructed are used in four failure prediction schemes among which a hybrid scheme achieves 91% predictability of failures with 99.3% coverage of prefixes in the Internet.**

I. Introduction

Autonomous Systems (ASes) in the Internet exchange routing information via the Border Gateway Protocol (BGP), by advertising paths to *prefixes* – which are aggregates of IP addresses. These paths can be withdrawn through routing updates, due to several reasons like link failures or AS policy changes, causing the prefixes to become unreachable from various portions of the Internet. The goal of this paper is to predict these routing failures. For each given prefix of interest, we determine its *BGP molecule*, which is a group of Internet prefixes with similar failure characteristics. BGP molecules generalize the concept of *BGP atoms* [1], which are prefix clusters such that all BGP routers (peers) which can reach prefixes in the same atom do so using the same AS paths. We consider similarity in AS paths to prefixes as just one possible metric in constructing molecules. The BGP molecules, unlike BGP atoms, can consist of prefixes belonging to different ASes, just like a molecule can consist of atoms belonging to different chemical elements.

While BGP atoms were introduced to aggregate BGP prefixes that are subject to the same policy [1], our goal in forming BGP molecules is formulating a fundamental unit which can be used in effective diagnosis of routing problems, ultimately improving the security and reliability of the Internet control plane. We study the potential of BGP molecules in predicting failure of the prefix of interest by considering four failure prediction algorithms, with and without the use of BGP molecules, in Section VI. We find that BGP molecules are easy to compute and achieve higher failure prediction accuracy than prediction schemes using BGP atoms.

We develop correlation coefficient metrics for comparing two prefixes in terms of their failure tendency (Section IV). We consider a number of prefix characteristics to determine their relationship with the correlation coefficients (Section V).

BGP molecules give us information about which prefixes are likely to be affected by a single event, which aids failure prediction and diagnosis algorithms. One can then develop a reactive routing mechanism to route around failures [2]. For instance, iPlane Nano [3] showed that intelligently selecting detours can improve the performance of routing. BGP molecules also reveal similarity in failure tendency and can be used to study how to improve the reliability of web-based applications and cloud computing, so that the primary and backup prefixes can be placed in different BGP molecules.

II. Background and Related Work

There has been a significant amount of research on diagnosis of BGP routing events [4], [5], [6]. Three dimensions are used for identifying the origin of a routing event: *time*, *prefixes*, and *views*. Wu *et al.* [5] point out that grouping updates across prefixes reduces the number of trouble reports sent to the operators and aids diagnosis. Prior work only safely correlates updates across prefixes when the number of prefixes updated exceeds a threshold, e.g., 100 [6]. Our work can enhance these diagnosis algorithms since we group prefixes by their failure tendencies using well-formed metrics.

Diagnosis is not the only goal of clustering prefixes; gaining a better understanding of the the prefix address space similarity is an important goal. Prior work [2], [7] has studied the correlation of data plane failures, measured through active probing, with BGP routing updates, for predicting the data plane reachability failures of prefixes. The prediction is done for one prefix at a time using its updates, and some prefixes/portions of the Internet are found to be more predictable than others. A better understanding of the similarity among prefixes in the control plane would likely improve predictability. It can also lead to a better selection of prefix candidates for further inspection by data plane monitoring systems like Hubble [8], and deeper insight into the behavior of subset and superset prefixes [9] in failure scenarios.

III. Data Sets

The routing tables and updates available from RouteViews [10] from March 2009 are used to build BGP molecules and for failure prediction. We selected this month since no

known major routing event (such as an undersea cable cut) occurred, in order to produce unbiased results. This is important because the BGP molecules will typically be used in normal operation scenarios, as significant routing events are rare. We define a peer as any vantage point in our dataset which is present in any routing table entry and at least one update. We executed scripts (contributed by the authors of [11]) from the point of view of each of the 42 peers in our dataset to remove spurious updates due to routing table transfers. We use these filtered updates for all further processing.

Each prefix announcement and each routing table entry is associated with an AS path and a peer. The origin AS of the prefix is the last AS on the AS path. We extracted 31,576 unique origin ASes out of the data visible for the month, and stored the prefixes that they originate along with an array of the times of prefix state changes. The state of a prefix can be Up (U) when the prefix is in an announced state or Down (D) when the prefix is in a withdrawn state. Each prefix has a state change array for each of the peers that can reach this prefix.

We found that about 1.6% of the prefixes exhibited MOAS conflicts, i.e., their origin could be attributed to two or more ASes. Since our focus in this paper is not on resolving MOAS conflicts, we attribute the prefix to all of the ASes that appear to originate it. We change the states of the prefix to "Down" for *all* the ASes that originate this prefix on seeing a withdrawal. The origin AS of about 0.013% of the prefixes is an AS_SET [12], which we keep as a separate entity. The number of prefixes originated by an AS ranges from 1 to 4402 in our data. We had 329,658 prefixes in our dataset with an average of 10.44 prefixes originated by an AS. However, the distribution is highly skewed with about 42% of the ASes originating only 1 prefix and about 86.2% of the prefixes originating less than or equal to 10 prefixes.

IV. Metrics

We now define the correlation coefficient based metrics, a high value of which indicates that the failure tendencies are close. For each prefix and each peer which can reach it, we have state change sequences of the prefix recording the time when the state of the prefix goes from U to D or vice versa. We compare the state change arrays of two prefixes when they are viewed by the same peer.

We define "failure correlation coefficient" as: $\frac{DD-DU-UD}{DD+DU+UD}$, where $\{$xy$\}$ with x, y = U or D denotes the total time in the month long dataset when the first prefix state is x and the second prefix state is y. In our dataset of 329,659 prefixes, prefixes are 'Up" for most of the time (92.37% of time on the average). Hence, we only consider the time when at least one of the prefixes has failed. This captures the correlation between the failure tendencies of two prefixes more accurately, since it evaluates whether one prefix has failed, given that the other prefix has failed. We study other correlation coefficient metrics in the extended version of this paper [13].

Unfortunately, computing the failure correlation coefficient of all prefix pairs in our dataset is a virtually impossible task since we have 329,658 prefixes and hence about 54.3 billion prefix pairs. Even if the correlation coefficient of a prefix pair takes one second to compute, it would take 1746 years to compute the coefficients for all the pairs on a single processor machine. Hence, we randomly choose 1% of the ASes (or 316 ASes) and only consider the 2353 prefixes originated by those ASes. As Figure 1 shows, the frequency of prefixes originated by an AS in the 316 AS random sample is about the same as that of the entire set of 31,576 ASes, indicating that the prefix sample is a representative one for the prefixes of the entire Internet. Although we use this reduced dataset for the remainder of the paper, the randomness of our selection process makes the results for correlation coefficients unbiased. Further, the construction of BGP molecules relies on finding *some* prefixes in the Internet which are similar in failure tendency to the prefix of interest. Limiting the sample only makes our BGP molecule construction and the subsequent prediction mechanism appear worse than if more powerful computation mechanisms had been available.

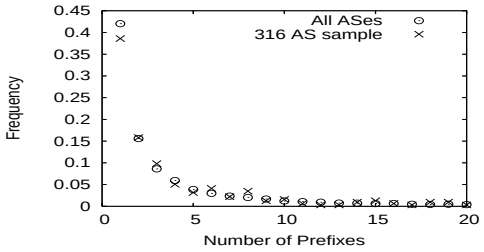

Fig. 1. Comparison of partial histograms of the number of prefixes in an AS in the random sample vs. for all ASes

The total number of prefix pairs for which failure correlation coefficients can be computed is 2.724 million vs. the 2.76 million pairs possible, as the pairs where both the prefixes were up for the entire month w.r.t. every peer are omitted. The failure correlation coefficient's median is -0.999999 with mean -0.927647. This is because two arbitrary prefixes in the Internet are unlikely to have high tendency to fail together unless they share common characteristics. It is our goal to find these characteristics. The results can be likened to the fact that two arbitrary chemical atoms in a large enough sample of atoms are unlikely to be the same and hence an average "similarity coefficient" would be close to -1.

V. Constructing BGP Molecules

We now construct each BGP molecule, i.e., the set of prefixes in the Internet similar to the prefix of interest, with the goal of failure prediction (which we demonstrate in Section VI). We choose the prefix of interest, one each from the 2353 prefixes of the 316 AS sample so that we do not bias our results towards a specific AS/prefix group. We study correlation of the prefix originating AS with its failure correlation coefficient in [13] and find that it correlates somewhat with the failure tendency. However, about 49.5% of the ASes have a negative failure coefficient, which is a driver for finding additional prefix characteristics that influence its failure tendency.

A. Using AS Paths

In this subsection, we study the hypothesis that prefixes which have similar AS paths from one or more vantage points are expected to have similar failure tendencies. This idea of using AS paths to group prefixes was initially proposed in [1] where prefixes were grouped into BGP atoms if they have the same AS paths from every visible default router. This definition necessitates that prefixes belonging to the same atom belong to the same AS, since the last AS on the AS path is the one which originates the prefix. We apply a broader definition defining "AS path sequences" as the AS paths occurring in the routing tables with the *first AS, last AS, and AS path prepending removed*. The first AS is of the vantage point which sees the prefix and is uninteresting when we aggregate data across vantage points, whereas the last AS is the originating AS of the prefix. AS path prepending [12] is removed since that has no implication on the sequence of ASes traversed between the vantage point and the prefix.

We form a "routing table set" containing the largest table for each day in the dataset. This eliminates short and possibly corrupted routing tables and improves computational efficiency. Routing tables closely spaced in time are expected to have significant overlap in their entries. We obtained 30 routing tables for March 2009, yielding a "combined routing table" with 14.8 million entries.

We narrow down the group of 2.76 million failure correlation coefficients for computational reasons by choosing sets of increasing coefficient values from 0 to 1 differing by at least 0.02 from the previous set and choosing no more than 1000 values for each set. This reduces our group to about 60,500 prefix pairs and for each of those, we see if we have AS paths for both of the prefixes in the pair from at least one peer in our combined routing table. We then compare the AS paths from the peers, one at a time, to compute "AS path correlation coefficient" (defined in the next paragraph). These coefficients are then averaged across peers to determine an AS path correlation coefficient for the prefix pair.

If the length of the AS path sequence for prefix 1 is l_1 and that of prefix 2 is l_2, we compute the Longest Common Subsequence (LCS) using the dynamic programming algorithm of [14], and define the AS path correlation coefficient as LCS/$min(l_1, l_2)$. Thus, the coefficient can range from 0 to 1.

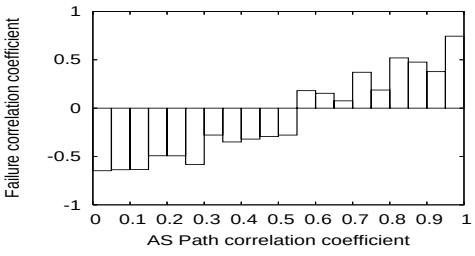

Fig. 2. Variation of average failure correlation coefficient of each bin with AS path correlation coefficient

We divide the AS path coefficients for the prefix pairs obtained into 20 bins and compute the average failure correlation coefficient of each bin, the results of which are shown in Figure 2. This shows a clear positive correlation between the two coefficients: as the AS path correlation coefficient between two prefixes increases, their average failure correlation coefficient changes from negative to positive, changing signs at an AS path coefficient = 0.55. This validates our hypothesis that AS path similarity is indeed a measure of prefix failure tendency.

To construct BGP molecules using AS path alone, we use only the first routing table since the goal is to use them for failure prediction. For each of the prefixes of interest, we find its AS path sequences w.r.t. each peer, and then search for other prefixes in the routing table which have the same AS path sequence w.r.t. the same peer and place them in its BGP molecule. We ignore AS path sequences which are just one AS long as that is too general a comparison.

B. On a Geographical Basis

We use MaxMind's GeoLiteCity application [15] to find the latitude and longitude of the location of the dotted decimal portion of the prefix, and then compute the distance between two prefixes of about 60,500 prefix pairs (Section V-A) using the Haversine Formula [16] for computing the great-circle distance. We find that out of prefixes at the same location, about 92.5% belong to the same AS, indicating that the geographical distance between prefixes is a different dimension from their originating ASes. Zero distance between prefixes does *not* imply that the prefixes belong to the same AS. The percentage of prefixes belonging to the same AS reduces to 90% for prefixes with distance less than 150 miles and to 70% for distance less than 600 miles.

We now evaluate whether geographical distance correlates with the failure correlation coefficient of prefixes. We have bins of 50 miles each, and we place each of the 60,500 prefix pairs into one of the bins depending on their distance. We then compute the average failure correlation coefficient of each bin. The results indicate that increasing distance corresponds to a lower similarity in failure tendency. The results for the first 600 miles [13] suggest that prefixes with distances 150 miles or less have a fairly high failure correlation coefficient, whereas those with greater distances have a negative coefficient.

C. Hybrid Scheme

From the above discussion, AS paths to a prefix are a stronger dimension than its geographical location in correlating with its failure tendency. However, there are several cases when AS paths alone do not yield any prefixes within a molecule. This may be because (i) the prefix of interest is not found in the routing table used, or (ii) the AS path sequences for finding similar prefixes are only one AS long, or (iii) there are no prefixes in the routing table with the same AS path sequences. Additionally, the number of prefixes in the molecule may be insufficient for prediction purposes (Section VI-B). We therefore devise a hybrid scheme for constructing BGP molecules. We find the prefixes which are

within a threshold distance (150 miles) of the prefix of interest, and place them in the molecule constructed using AS paths.

VI. Predicting Failures

A. Failure Prediction Methodology

Our prediction methodology involves failure prediction of prefixes of interest given "similar" prefixes in some regard. We select a set of 25 random "similar" prefixes for prediction purposes. The number 25 was selected to be large enough to give a meaningful sample, but small enough for low computational overhead in an online prediction application. If the number of "similar" prefixes is less than 25, we typically do not perform prediction except for evaluation reasons. Generally, a failure of the prefix of interest is predicted if a majority of the 25 prefixes fail during a time window, which is kept as a parameter. This prediction application can be easily deployed in the real world if failures of prefixes can be observed (e.g., through a live update feed). The use case of ISPs will typically have such a feed through peering, else they can be obtained from a public source like RouteViews [10]. We execute our prediction experiments for all 2353 prefixes.

We now present an example, with the prefix of interest 210.143.240.0/20. Figure 3 shows the indices of 25 random prefixes in its molecule (numbered 0 to 24) and their failure time. Since 13 prefixes fail within 1 second of each other ($<$ time window t=300 seconds), we predict that the prefix of interest will fail within t seconds beginning at t_0. Since it failed at t_0+1 within t seconds of the failure of the first prefix out of 13 prefixes, we consider this failure predictable.

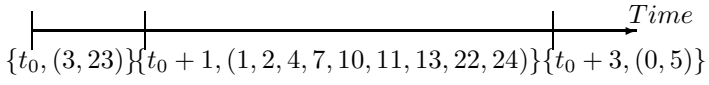

Fig. 3. Example of failure prediction using BGP molecules, t_0=1235877308 Unix time, Each label has {time,(list of prefix indices which fail at that time)}

B. Evaluating Prediction Quality

Let F denote the failure event of a prefix. We formulate the following hypotheses. Null Hypothesis H_0: F happens within a time window t when the application predicts a failure. The alternative hypothesis H_1 states the case that F does not happen within t given the application predicts a failure. Any evidence in support of the null hypothesis favors the success of our prediction application. We do not require exact time synchronization, since we evaluate the feasibility of the prediction application; our approach is similar to that in [2].

We form the likelihood ratio:

$$\Lambda = \frac{P(H_1 \; is \; true)}{P(H_0 \; is \; true)} = \frac{P(No \; F \; within \; t \; | Application \; predicts \; F)}{P(F \; within \; t \; | Application \; predicts \; F)}$$

A large value of the likelihood ratio indicates that the alternative is true; hence we reject the null when $\Lambda > \gamma$ where γ is decided by using two disjoint but randomly selected sets, namely training and test sets of prefixes which are "similar" to the prefix of interest. These sets usually have 25 prefixes, unless specified otherwise. We use the training set to find the value of γ by counting the number of instances when the alternative is true and dividing it by the number of instances where the null is true. However, γ is chosen to be at least 1, because we do not want to reject the null unless the evidence in favor of the alternative exceeds that of the null. After the value of γ is decided, we execute the same algorithm for the test set, compute Λ and reject the null if $\Lambda > \gamma$. Due to the inherent randomness in selecting the training and test sets, we perform five predictions for each prefix of interest, with different random seeds based on the current wall time.

It may not be possible to perform failure prediction for all prefixes, for example because of an empty BGP molecule. Hence, we define *coverage* of the prediction mechanism to be the percentage of the 2353 prefixes for which a decision on predictability can be made. Out of the prefixes for which prediction is possible, the prediction methodology is either successful or unsuccessful in predicting the failures of the prefix of interest. We define *predictability* as the percentage of prefixes whose failures are predictable.

C. Naïve Prediction

We first study a Naïve prediction model which does not use BGP molecules or any prefix characteristics for failure prediction. It learns other prefixes that fail with the prefix of interest during a learning duration, and uses these prefixes to predict failure. It is computationally intensive: For each prefix of interest among the 2353 prefix sample, we identify prefixes in the sample, which fail within a time window of its failure during a day-long learning duration (March 1^{st}, 2009). The window is selected to be 300 seconds to allow sufficient time for routing convergence, which has a median time of about 3 minutes [17], delayed update visibility and time synchronization issues. We find the predictability to be 80.6%. While this is promising, the high computational complexity of this prediction method makes it infeasible.

D. Using BGP Atoms

For each of the 2353 prefixes of interest, we compute the set of prefixes that are in the same BGP atom [1]. The primary disadvantage of this scheme is that for most cases, we do not find any prefixes in the same atom since they also have to be in the same AS. About 42% of the ASes have only one prefix (Section III). Even if an AS has multiple prefixes, it is difficult to find prefixes in the same BGP atom because multiple prefixes may be advertised with different policies for load balancing, leading to different AS paths.

The average number of prefixes in non-zero-sized BGP atoms is 2.88 vs 11.41 for molecules formed using AS paths. The maximum number of prefixes in a BGP atom is 23 which is less than 50. Thus, we randomly assign about half of the prefixes to the training and test sets in equal numbers, when we have at least 2 prefixes in the atom. This still only gives us a coverage of 1.66%, and a predictability of 87.2%.

E. Using BGP Molecules Constructed by AS Paths

We now consider prefix failure prediction using BGP molecules constructed using AS paths as in Section V-A. Table I compares the performance of this AS path-based prediction with the two other prediction schemes studied so far in terms of failure predictability and coverage. The results show that using BGP molecules is the best prediction scheme studied so far with about 14% higher predictability than Naïve prediction. However, it still suffers from the disadvantage that slightly less than half of the prefixes are predictable. Hence, we study the hybrid prediction scheme in the next section.

TABLE I
FAILURE PREDICTABILITY PERFORMANCE OF BGP MOLECULES CONSTRUCTED USING THREE SCHEMES; FAILURE PREDICTION WINDOW=300 SECONDS

Scheme	Failure Predictability (%)	Coverage (%)	Disadvantage
Naïve Prediction	80.62	100	Computationally Intensive
BGP Atoms	87.2	1.66	Low # of Coverage
BGP molecules (AS paths)	91.82	47.2	Moderate Coverage

F. Using BGP Molecules Constructed by Hybrid Scheme

To improve coverage, i.e., percentage of prefixes for which a prediction can be made, we use a hybrid scheme (Section V-C). The "hybrid prediction scheme" operates as follows: (1) Predict failures of the prefix of interest using BGP molecules constructed using AS paths (Section V-A) if they have at least 50 prefixes. (2) Otherwise, construct molecules using geographical proximity and combine with the AS path molecule to form a hybrid molecule. Predict using the hybrid molecule if it has at least 50 prefixes.

TABLE II
PREDICTION RESULTS OF HYBRID PREDICTION SCHEME

Description	Number of prefixes	Coverage (%)	Failure Predictability (%)
AS path molecules do not have any prefix	1079	45.86 %	-
AS path molecules have < 50 prefixes	164	6.97 %	-
"Hybrid" molecules having ≥ 50 prefixes	782	33.23 %	93.58
"Hybrid" molecules having < 50 prefixes	461	19.59 %	83.51
Hybrid prediction combining all techniques	2336	99.28 %	90.83
"Hybrid" molecules having < 2 prefixes	17	0.72 %	-

Table II gives the coverage and predictability of this prediction scheme. For the 461 cases where hybrid molecules have at least 2 but < 50 prefixes, we divide the prefixes into two equal parts of training and test sets. The overall hybrid scheme is the best. It has a high coverage of 99.28% excluding only 17 prefixes of interest and a predictability of about 91%.

VII. CONCLUSIONS AND FUTURE WORK

This work has focused on using prefix characteristics to construct a group of prefixes similar in failure tendency to a prefix of interest (called a "BGP molecule") with the primary goal of predicting failures. To the best of our knowledge, this is the first work which has evaluated the similarity of prefixes in the Internet w.r.t. their failure tendency and shown its feasibility for prediction applications. We evaluate four schemes to predict failures and show that a hybrid scheme based on AS paths and geographical location performs the best with 91% predictability and 99.3% coverage.

As future work, we plan to develop an online tool for predicting control plane reachability failures, and consider prefixes that are more specific versions of other prefixes. Finally, we plan to use our tool to further study the interplay of data plane and control plane reachability.

REFERENCES

[1] A. Broido and K. Claffy, "Analysis of RouteViews BGP data: policy atoms," in *Network-Related Data Management (NRDM) workshop*, 2001.

[2] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek, "Measuring the effects of internet path faults on reactive routing," in *SIGMETRICS '03: Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2003, pp. 126–137.

[3] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. . Krishnamurthy, and A. Venkataramani, "iPlane Nano: path prediction for peer-to-peer applications," in *NSDI'09*, 2009, pp. 137–152.

[4] M. Caesar, L. Subramanian, and R. H. Katz, "Towards Localizing Root Causes of BGP Dynamics," UC Berkeley, Tech. Rep. UCB/CSD-04-1302, 2003.

[5] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, "Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network," in *Proc. of NSDI*, 2005.

[6] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet Routing Instabilities," in *Proc. of ACM SIGCOMM*, 2004.

[7] Y. Zhang, Z. M. Mao, and J. Wang, "A framework for measuring and predicting the impact of routing changes," in *INFOCOM*, 2007, pp. 339–347.

[8] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson, "Studying blackholes in the Internet with hubble," in *Proc. of NSDI*, 2008.

[9] A. Broido, E. Nemeth, and K. Claffy, "Internet expansion, refinement and churn," in *European Transactions on Telecommunications*, 2002.

[10] University of Oregon, "Route Views Project," http://www.routeviews.org/.

[11] B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang, "Identifying BGP routing table transfers," in *Proc. of ACM MineNet workshop*, 2005.

[12] Y. Rekhter and T. Li, "RFC 1771 (BGP version 4)."

[13] R. Khosla, S. Fahmy, and Y. C. Hu, "BGP Molecules: Understanding and Predicting Prefix Failures," Purdue University, Tech Report, 2011, available: http://www.cs.purdue.edu/homes/rkhosla/protected_post/techreport_infocom11.pdf.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[15] MaxMind, "GeoLite City," http://www.maxmind.com/app/geolitecity.

[16] R. Sinnott, "Virtues of the haversine," *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984.

[17] S. Burkle, "BGP convergence analysis," Ph.D. dissertation, 2003.