Predicting Prefix Availability in the Internet

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

Abstract—**The Border Gateway Protocol (BGP) maintains inter-domain routing information by announcing and withdrawing IP prefixes, which may result in prefix unreachability. Prefix availability observed from different vantage points in the Internet can be lower than standards promised by Service Level Agreements (SLAs).**

In this paper, we develop a framework for predicting long-term prefix availability, given short-duration prefix information from publicly available BGP routing databases. We compare three prediction models, and find that bagged decision trees perform the best when predicting for long future durations, whereas a simple model works well for short prediction durations. We show that mean time to failure and recovery outperform past availability in terms of their importance for predicting availability for long durations, and predictability is higher in the year 2009, compared to four years earlier. Our models allow ISPs to adjust BGP routing policies if predicted availability is low, and the models are useful for cloud computing systems, P2P, and VoIP applications.

I. Introduction

The Border Gateway Protocol (BGP), the de-facto Internet inter-domain routing protocol, propagates reachability information by announcing paths to *prefixes*. Autonomous Systems (ASes) maintain these paths to prefixes in their routing tables, and conditionally update this information when route update messages (announcements and withdrawals) are received. Continuous prefix reachability over time is critical to the smooth operation of the Internet. This is captured using the metric of *availability*, defined as the time duration when the prefix is deemed reachable divided by the total time duration we are interested in. While typical system availability metrics for telephone networks exceed *five 9s,* i.e., 99.999%, computer networks are known to have lower availability [16].

Internet Service Providers (ISPs) usually provide availability guarantees on their backbone network through Service Level Agreements (SLAs). However, content providers are more interested in their service availability as observed from various points in the Internet, and a routing path being advertised is essential to maintaining traffic flow. There have been unsuccessful attempts at extending SLAs to several ISPs [14] or estimating service availability between two end points [8]. Meanwhile, several Internet reachability problems have been reported, such as the YouTube prefix hijack which lasted about two hours [20], and several undersea cable cuts, e.g., [2].

Measuring prefix availability is non-trivial without an extensive measurement infrastructure comprising many vantage points. *Data plane* measurements are inherently discontinuous and increase network traffic. A shortfall in measured availability necessitates a *reactive* approach to correct the problem after the fact.

This paper takes a *predictive* approach towards the long-term *control plane*, BGP-advertised availability of prefixes from several vantage points in the Internet. We construct models for predicting availability, and show that, given a long-enough *learning duration* for a prefix, we can simply predict its future availability to be equal to its past availability (Section IV-A). However, if the learning duration is short compared to the prediction duration, we use data-mining based prediction models constructed using routing information of other Internet prefixes (e.g., from RouteViews [15]) to predict prefix availability. A predicted long-term availability value which falls short of requirements could lead to changes in BGP policies of the ISP regulating the prefix advertisements. For example, one can change the MED or community attributes, or increase the penalty threshold associated with route flap damping to a high availability requirement prefix to ensure fewer flaps [3]. Additional applications of our work include Content Distribution Networks (CDNs), cloud computing, VoIP and P2P networks, all of which can use the highest predicted availability replica/server/peer.

Internet prefix characteristics can convey valuable information about other (not necessarily in the same AS) prefixes. We use randomly selected prefixes from RouteViews to build data-mining based prediction models, which are used to predict availability of unrelated prefixes. This theme is common in several other disciplines, like medicine, where one uses known symptoms of patients with a diagnosed disease to try to diagnose patients with an unknown condition. To the best of our knowledge, no other work has exploited the similarity of prefixes in the Internet; a few studies, e.g., [18], applied predictive modeling in the context of BGP, but they only examined problem ASes in the path to a particular prefix.

The remainder of this paper is organized as follows. We define the problem that we study in Section II. Section III describes our methodology. We evaluate and compare three prediction models in Section IV. Section V summarizes related work, and Section VI concludes the paper.

II. Problem Definition

We define the *availability prediction problem* to be the prediction of the BGP-advertised availability of a prefix, given its attributes computed by observing BGP updates for the learning duration, and the availability and attributes of other prefixes, which are used if needed.

In this paper, we compute availability in the *control plane* from a particular vantage point by marking the time of an

R. Khosla, S. Fahmy, C. Hu, and J. Neville. Predicting prefix availability in the internet. In Proceedings of the 29th IEEE Conference on Information Communications (INFOCOM'10), pp. 216-220, 2010.

announcement/withdrawal of a prefix as the time when it goes up/down w.r.t. the vantage point and matching our predictions against this computed availability. Instead of predicting continuous availability values, we discretize availability, and predict the availability *class* of a prefix. This is because, for diagnosis or detection purposes, our interest lies in predicting whether the availability value is above or below an acceptable threshold (e.g., that advertised in an SLA), and not the specific value of the availability. Discretizing gives us the added advantage of using confusion matrix-based measures, e.g., false positives, to assess prediction performance.

We seek answers to the following questions for our framework:

1) How to discretize availability? How many classes and what threshold values should be used?
2) Given a set of prefixes with associated attributes and availability classes, how accurately can one predict the availability classes of other prefixes, and which prediction models work best?
3) Which attributes of a prefix are most important in predicting availability?
4) How long should one observe prefix attributes so that its availability can be accurately inferred?

III. Methodology

A. Datasets

We utilize routing tables and updates from RouteViews [15] for the months of January 2005, January 2007, February 2008, and March 2009 to build and test our prediction models. The months span a reasonable time period to prevent biasing our model selection process towards datasets from a particular timeframe when some event (e.g., undersea cable cut) may have occurred. We filter routing table transfers as described in [9]. In most of this paper, we study one month of data at a time, with the task of predicting the availability of combinations observed in the month using attributes observed in the first 25% of the month, i.e., about one week (we vary the learning duration in Section IV-E).

B. Defining and Discretizing Availability

For the RouteViews data, the vantage points w.r.t. whom the prefix availability is computed, are the RouteViews peers. We define a *combination* as a (peer, prefix) tuple, indicating that the prefix was observed by the peer in the dataset. We compute attributes of these combinations and use that for building and evaluating our prediction models. In what follows, a combination is *up* or *down* when the peer associated with the combination has the corresponding prefix in an announced or withdrawn state respectively.

The computation of the availability of a combination for a particular time period proceeds as follows. The first routing table of the period is used to initialize the state of each combination present in the table to *up* (U). We maintain the state of each combination at each point in time, and at the time of each state change due to an update, we record a downtime or an uptime depending on the state change. After processing all the update files, we add an extra up or downtime depending upon the last state of the combination. For example, if the last state change was to D and was reported at time t_1, and if the data period ended at time t_2, as indicated by the timestamp of the last update, we add a downtime with value $t_2 - t_1$. The availability of the combination is computed by dividing the total time that the combination was up by the total time period, only if it has at least one recorded uptime or downtime so that its availability is non-trivial.

We discretize the continuous availability value into two availability classes, namely *High* and *Low* using a single threshold of 0.99999, which leads to percentage of *High* combinations as 56.10%, 60.76%, 62.17% and 68.75% for the months of Jan. 05, Jan. 07, Feb. 08 and Mar. 09 respectively.

C. Model Space and Performance Metrics

We investigate three prediction models: a simple baseline prediction model, and two data-mining based prediction models, namely Naïve Bayes, and decision trees (with and without bagging) [17], as presented in [9]. The performance of each prediction model is studied using *n-fold incremental cross-validation* [17], with *n* = 10 and applied *k* = 10 times.

We use the confusion matrix-based measures of Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) and Kappa statistic [17] to evaluate the performance of prediction models. Unfortunately, confusion matrix-based measures can be misleading with a skewed class distribution. A better metric is obtained by using Receiver Operating Characteristic (ROC) curves [17], [5], which plot the TPR versus the FPR and are independent of class skew. We use the Area Under the ROC Curve (AUC) as a performance metric. which is the probability that the model will rank a randomly chosen *high* instance higher than a randomly chosen *low* instance. A purely random classifier (which randomly selects the class label) has an AUC of 0.5, whereas a perfect classifier has an AUC of 1.

IV. Prediction Models

A. Simple Prediction

We first consider a simple model, which does not learn based on other combinations, but merely predicts the future availability for a combination to be the same as its *past availability*, which is its availability during the learning duration. Thus, if a combination's *past availability* exceeds 99.999%, the predicted class label is *high*, otherwise it is *low*.

The performance metrics computed for the simple model for the four months of data and averaged over *nk* = 50 runs, are listed in Table I. The results show that while the TPR of the simple model is high, its FPR is high as well. However, the simple classifier outperforms a random classifier (as indicated by the κ statistic) and hence forms a baseline model to which other sophisticated models can be compared.

This model does not rank instances in terms of probabilities of being classified as *high*/*low*, and hence produces a single point in the ROC space. For computing ROC based metrics, we take a typical run of the model with confusion matrix measures close to their average values. The instances which are classified

TABLE I
RESULTS WITH THE SIMPLE PREDICTION MODEL

Month	Accuracy (%)	TPR	FPR	κ	AUC
Jan. 05	67.68	0.9946	0.7195	0.2959	0.6319
Jan. 07	72.08	0.9961	0.6905	0.3444	0.6223
Feb. 08	77.97	0.9971	0.5704	0.4778	0.7076
Mar. 09	83.34	0.9977	0.5466	0.5327	0.7208

as *high* and *low* are randomly reordered within their respective groups, and then are ranked with the (predicted) *high*s higher than the *low*s. We now vary the prediction threshold, as in Algorithm 2 of [5] to compute the points on a ROC curve. The AUC is computed, using Algorithm 3 of [5], for the typical run and averaged across 50 runs.

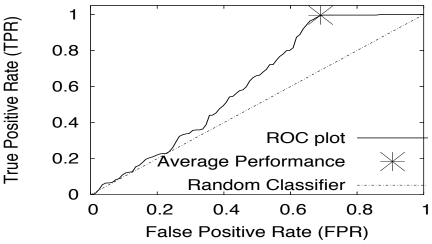


Fig. 1. ROC plots for the simple prediction model for Jan 07

The ROC curve for the simple prediction model for a typical month is shown in Fig. 1. The plots show the original model performance (in Table I) as a point ("star") on the ROC plots, along with the performance of a random classifier. The performance of simple prediction is clearly better than a random classifier, but there are occasions when it performs as good as a random one. Recent months show better performance, with a TPR close to 1 being reached for a smaller FPR.

B. Computing Attributes

In this section, we study the attributes of a combination, which are computed from RouteViews data and are used to train data-mining based prediction models. They are selected to relate to its availability, and to be easily computable so that the learning system is fast. It is important to note that these attributes do not necessarily cause high/low availability; we are looking for correlation *not causality*. Correlation is sufficient for a prediction model to be successful.

We compute the following attributes of each combination for the learning period: (1) Prefix length, (2) Update frequency, (3) Mean Time to Failure (MTTF), and (4) Mean Time to Recovery (MTTR). We investigate these attributes further in [9], and show that these attributes are indeed statistically significantly different between the *High* and *Low* classes. Although we compute the attributes of every combination with at least one recorded uptime or downtime, we downsample this set of combinations randomly to a set of 5000 combinations with their attributes, and use that to build and test models.

C. Decision Trees

We implement Decision trees using C4.5 algorithm using Weka [17], an open-source data mining software and use Reduced Error Pruning (REP) as the pruning technique. The high variance of decision trees can be reduced by *bootstrap aggregating (bagging)* [17].

We apply the Bagged Decision Tree classifier to predict availability for the four months of data, with the results presented in Table II. We perform the Welch t-test [10] to test for statistical significance w.r.t. the simple model with randomization and find that the average AUC performance increase of 9.03% is significant at 1% significance level, whereas the accuracy changes are not significant. Hence, we conclude that Bagged Decision Trees outperform the simple model with a higher ranking quality (AUC) and no significant change in accuracy. We also apply bagged decision tree models using each of Jan. 05, Jan. 07, and Feb. 08 training data to predict Mar. 09 availability. We find that no AUC changes were significant at 5% significance level, though a significant reduction in accuracy was observed when using Jan. 05 and Jan. 07 to predict Mar. 09 results. This shows that one can apply models trained with data from a year earlier with no significant performance degradation, facilitating adoption into a prediction infrastructure.

TABLE II
RESULTS WITH BAGGED DECISION TREES (% CHANGE FROM SIMPLE MODEL GIVEN WITHIN PARENTHESES)

Month	Accuracy (%)	TPR	FPR	AUC
Jan. 05	67.83 (0.23%)	0.9616 (-3.32%)	0.6746 (-6.24%)	0.7005 (10.86%)
Jan. 07	72.50 (0.58%)	0.9779 (-1.83%)	0.6530 (-5.44%)	0.7094 (14.00)%)
Feb. 08	77.80 (-0.22%)	0.9927 (-0.44%)	0.5682 (-0.39%)	0.7483 (5.75%)
Mar. 09	83.24 (-0.12%)	0.9976 (-0.01%)	0.5501 (0.64%)	0.7605 (5.51%)

D. Classification Attributes

We now explore the importance of the attributes used in prediction. We start with the results from Table II, and remove certain attributes of the combinations, which are fed to the Bagged Decision Tree model. The degradation in various performance metrics is studied; as degradation increases, the importance of the removed attribute subsets also increases. We present typical results of removal of some of the attributes in Table III. The first column of the table indicates which attributes of the combinations were used for prediction. For comparison, we build a decision tree model using only *past availability*, which was used in the simple prediction model.

We conclude that the performance degrades significantly when only past availability is used as a single attribute with lower AUC (5-9%) and significantly higher FPR (6-11%), which together with the simple model results, implies that past availability is not adequate for prediction of future availability when predicting for future durations longer than the learning duration. Prefix length and update frequency

TABLE III
PERCENTAGE CHANGES IN PERFORMANCE METRICS WITH SUBSETS OF ATTRIBUTES FOR MAR. 09. ALL PERCENTAGE CHANGES ARE W.R.T. MAR. 09 RESULTS IN TABLE II

Attributes used for prediction	% change in accuracy	% change in TPR	% change in FPR	% change in AUC
Past availability	-2.03	0.24	11.11	-5.33
Prefix length, Update frequency	-10.37	-9.83	10.71	-6.63
MTTF	-0.85	-1.54	-2.23	-4.40
MTTR	-0.005	0.01	0.07	-4.63
MTTR, Prefix length Update frequency	0.03	0.02	-0.05	-0.35
MTTF, Prefix length Update frequency	-0.88	-1.52	-1.97	0.09

are weaker attributes since using them causes the AUC to decline by 4-8%. MTTF & MTTR are most important since their use alone causes the least drop in AUC among any single item attribute set. We also experimented with adding *past availability* to these attribute subsets and found that the performance did not change significantly. It is worth noting that the prediction model selects MTTF and MTTR as the most important attributes for predicting long-term availability, showing that they shed valuable insights into the future.

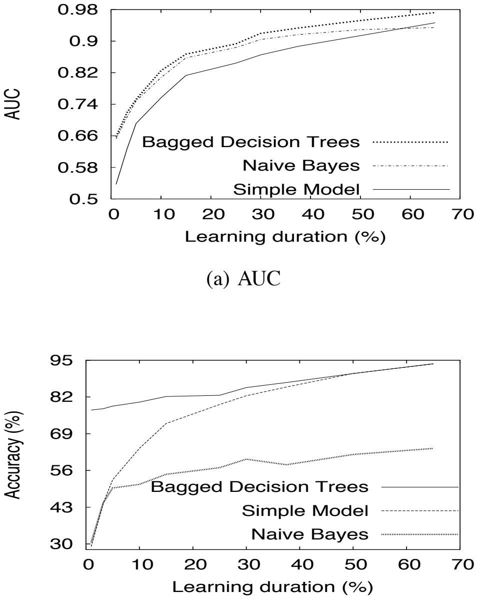
E. Learning Duration

Lowering the learning duration (thus far at 25%) will lead to a deterioration in prediction results, since we have less information for prediction, and increasing it has the opposite effect. We study this effect on a 12 months prediction duration (January to December 2007). The variation of AUC and accuracy versus the learning duration percentage is shown in Fig. 2. It is surprising to note that the accuracy for Naïve Bayes and the simple model is extremely low for lower percentage durations, most likely because they are not ensemble predictors like bagged decision trees. The decision tree model performs very well compared to the other two, especially at short learning durations (which we are most interested in for longer duration predictions), with accuracy never falling below 77%.

With a 5% learning duration, the accuracy and AUC for the decision tree model are 75.5-79% and 0.69-0.75 respectively. If these performance levels are acceptable, one can predict availability for about 20 times the learning duration. If we require about a 90% accuracy and about 90% AUC we must learn from about 50% of the duration at which point a simple model is preferred for its simplicity. Our prediction framework allows the system administrator to trade off accuracy and prediction duration with the model complexity.

F. Discussion

A comparison of prediction models on the four months of data with 25% learning duration is depicted in Fig. 3. The results show that Bagged Decision Trees perform best among the models considered for all four datasets as they have the highest AUC, a 9% average AUC gain over the simple prediction model, with about the same accuracy.



(a) AUC

(b) Accuracy

Fig. 2. Effect of learning duration on prediction performance for Jan.-Dec. 07

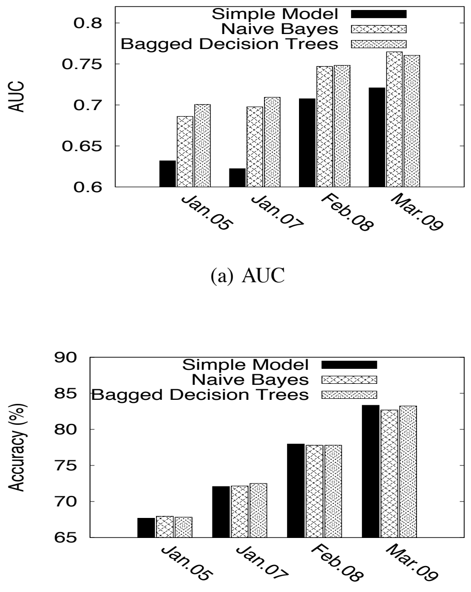
Bagged Decision Trees also perform the best for low learning duration percentages in terms of both accuracy and AUC (Fig. 2). At lower learning duration percentages, they have a much higher accuracy than other models and an AUC that matches that of Naïve Bayes. At higher percentages, i.e. when predicting for a future duration which is about the same as the learning duration, we would prefer the simple model for its simplicity over a data mining based model. Lower percentages pose a harder prediction problem which is best tackled by a data mining based solution.

The results in Fig. 3 suggest that for more recent months, all prediction models perform better in terms of both accuracy and AUC. The reason is that the behavior of the combinations over the learning period becomes a better indicator of their future performance. We can consider this to be a measure of "Internet health" because a more predictable Internet can aid in fault diagnosis. This new dimension of Internet health is an addition to observations made by Li *et al.* [11].

V. RELATED WORK

Prefix attributes like activity, update count, reachability from various monitors, prefix churn, and growth, have been studied, e.g., in [1], [19], [13], [6], but the attributes are not used to classify prefixes or predict prefix features, as in this paper.

Chang *et al.* [4] cluster routing updates into events based on the announcing peers and AS path similarity using *descriptive modeling* as the data mining technique, which is used for understanding the data. In contrast, we use *predictive modeling* to predict prefix behavior, specifically availability, given the observed prefix attributes and a learned prediction model.



(a) AUC

(b) Accuracy

Fig. 3. Results for four months.

Zhang *et al*. [18] predict the impact of routing changes on the data plane. They aim to predict reachability problems based on problematic ASes in AS paths in the routing updates. Our work is orthogonal to theirs in the sense that we consider control plane availability, utilizing four simple attributes computed from RouteViews data, and we investigate three prediction models that learn from other prefixes. Recently, Hubble [7] and iPlane [12] have been developed to detect data plane reachability problems and predict data plane path properties, respectively. Our work is complementary to theirs since we predict control plane availability (or existence of routing paths) to a prefix from multiple Internet vantage points, while they sample data plane metrics like latency, bandwidth and loss rates to end hosts at a low frequency. Our work can be combined with theirs to improve the performance of several applications like VoIP, P2P, and CDNs. In cases where no responsive hosts within a prefix can be found by iPlane, it cannot make predictions, in which case our availability predictions will be the only ones available for applications.

VI. Conclusions and Future Work

In this paper, we have developed a long-term availability prediction framework which uses a simple prediction model to predict future availability for a duration lesser or equal to the learning duration, while learning a prediction model from a set of Internet prefixes, for longer prediction durations.

We show that Bagged Decision Tree (BDT) model, which uses four easily computable prefix attributes, performs the best for those longer durations, with a 9% average gain in AUC over a simple model. Longer durations pose a more difficult prediction problem, thereby making data-mining based models like BDTs the preferred choice. We learn these models using the attributes of a random set of Internet prefixes. We find that mean time to recovery or failure are the most important attributes for prediction, and past availability is not a good indicator of future availability for hard prediction problems. To the best of our knowledge, this is the first work that uses the similarity of prefix behavior in the Internet to predict properties such as availability.

We plan to extend our framework to predict availability of an arbitrary end point as viewed by an arbitrary vantage point by using techniques similar to [12]. We will also investigate additional prefix attributes, such as the ASes to which the prefixes belong, and the AS paths to the prefixes. Finally, we will rigorously compare control plane to data plane availability.

References

[1] Andre Broido, Evi Nemeth, and Kc Claffy. Internet Expansion, Refinement and Churn. In *European Trans. on Telecommunications*, 2002.

[2] Cable News Network (CNN). Internet failure hits two continents. http://edition.cnn.com/2008/WORLD/meast/02/01/internet.outage/index.html, January 2008.

[3] M. Caesar and J. Rexford. BGP routing policies in ISP networks. *IEEE Network*, 19(6):5–11, Nov.-Dec. 2005.

[4] Di-Fa Chang, Ramesh Govindan, and John Heidemann. The temporal and topological characteristics of BGP path changes. In *IEEE ICNP*, page 190, 2003.

[5] Tom Fawcett. ROC graphs: Notes and practical considerations for researchers. Tech Report HPL-2003-4, HP Laboratories, 2003. Available: http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf.

[6] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. Locating Internet Routing Instabilities. In *ACM SIGCOMM*, 2004.

[7] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying blackholes in the Internet with Hubble. In *NSDI*, 2008.

[8] R. Keralapura, C.N. Chuah, G. Iannaccone, and S. Bhattacharyya. Service availability: a new approach to characterize IP backbone topologies. *IWQOS 2004*, pages 232–241, June 2004.

[9] Ravish Khosla, Sonia Fahmy, Y. Charlie Hu, and Jennifer Neville. Predicting prefix availability in the Internet. Tech Report, Purdue University, 2009. Available: http://www.cs.purdue.edu/homes/rkhosla/protected_post/techreport.pdf.

[10] Erich Leo Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, New York, 3rd edition, 2005.

[11] Jun Li, Michael Guidero, Zhen Wu, Eric Purpus, and Toby Ehrenkranz. BGP Routing Dynamics Revisited. *SIGCOMM CCR*, 37(2):5–16, 2007.

[12] H. V. Madhyastha, T. Isdal, M. Piatek, . Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *OSDI*, pages 367–380, Nov. 2006.

[13] Ricardo V. Oliveira, Rafit Izhak-Ratzin, Beichuan Zhang, and Lixia Zhang. Measurement of highly active prefixes in BGP. In *GLOBECOM*, volume 2, 2005.

[14] Panita Pongpaibool and Hyong S. Kim. Providing end-to-end service level agreements across multiple ISP networks. *Computer Networks*, 46(1):3–18, 2004.

[15] University of Oregon. Route Views Project. http://www.routeviews.org/.

[16] Warren Montgomery. So you want 5 9's? http://home.att.net/~wamontgomery/communications/99999.htm.

[17] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. 2nd edition, 2005.

[18] Ying Zhang, Zhuoqing Morley Mao, and Jia Wang. A framework for measuring and predicting the impact of routing changes. In *INFOCOM*, pages 339–347, 2007.

[19] Xiaoliang Zhao, Dan Massey, S. Felix Wu, Mohit Lad, Dan Pei, Lan Wang, and Lixia Zhang. Understanding BGP behavior through a study of DoD prefixes. In *DARPA Information Survivability Conference and Exposition*, April, 2003.

[20] Earl Zmijewski. Threats to Internet routing and global connectivity. In *20th Annual FIRST Conference*, 2008.