Calculating Ordered AS-level BGP Routing Trees Efficiently

Cynthia Freeman
University of New Mexico
cynthiaw2004@gmail.com

Benjamin Edwards *
IBM Research
benjamin.edwards@ibm.com

Jedidiah R. Crandall
University of New Mexico
crandall@cs.unm.edu

Stephanie Forrest
University of New Mexico and
Santa Fe Institute
forrest@cs.unm.edu

ABSTRACT

Although the Autonomous Systems (AS) graph has been studied for years, the community lacks well-supported tools for Internet-scale analysis of routing paths. In this paper, we focus on understanding AS topology from the perspective of BGP paths. Since BGP routing is not deterministic, but based on local preferences at each AS, routing information is non-trivial to collect, and researchers typically rely on simulation to generate realistic paths. We are particularly interested in questions about large-scale properties of the Internet and how they enable or are affected by geo-political pressures such as censorship, surveillance, or data-localization rules. To address these kinds of questions requires knowing how data travel from a source to a destination AS for all pairs of ASes. In this paper, we present BGPSimPy, a simulation tool for inferring likely routes between all pairs of ASes. BGPSimPy builds on earlier work by using well-known and accepted heuristics for computing likely paths between ASes, and extending the functionality and scalability of BGPSim [13]. We present BGPSimPy, a tool that computes routing trees for all ASes and makes them publicly available, so users can compute likely routes between any pair of ASes on commodity hardware in minutes. We illustrate the value of BGPSimPy in the context of censorship, defining and investigating the *chokepoint potential* of 182 countries. This task requires knowing routes (ordered paths) for a large number of AS pairs, highlighting new functionality and scalability. Using these data, we consider correlations between a country's chokepoint potential and its censorship levels, revealing the power of BGPSimPy to identify large-scale patterns and trends in interdomain routing.

1. INTRODUCTION

Many studies of the Internet's interdomain routing system use modeling and simulation to investigate topics such as BGP convergence [16], geopolitical control of network traffic [6, 8, 31], and network reliability [33]. This work typically considers the graph formed by Autonomous System (AS) nodes where edges connect ASes with either peer-to-peer (no money is paid) or provider-to-customer (customer pays provider) links. A route is simply an ordered list of the ASes that a packet traverses from the source to the destination AS. Computing likely routes for the entire AS graph is a well-known challenge, because many routing tables are not public, routes do not necessarily correspond to shortest paths (there is incentive to prefer the cheapest routes, and political reasons to affect routing), and because of the sheer number of AS pairs.

BGPSim is an important tool that simulates likely routes, given source and destination ASes [13]. It computes a routing tree for any given destination AS, which consists of all possible paths to that AS. In this paper, we address several limitations of BGPSim: (1) BGPSim returns a set of ASes for each path it considers but not the order in which they are visited; (2) Once routing trees are determined, they cannot be accessed latter without recalculation; (3) BGPSim relies on outdated parallelization frameworks (DryadLinq).

Our tool, BGPSimPy, builds on the BGPSim approach and addresses the limitations identified above. It is implemented in Python and is freely available at `https://github.com/cynthiaw2004/BGPSimPy`. Using BGPSimPy, we created routing trees for all ASes listed in [9] and computed paths between all ASes using these routing trees over the course of 5 days with a mean wall clock time of .0086 seconds per path. 50,979 routing trees totaling 50 GB uncompressed are freely available for download at `http://www.cs.unm.edu/~freemanc/bgpsimpy/`.

We illustrate the value of these data by first discussing statistical properties of routing trees and paths across the entire Internet, which, e.g., can shed light on properties like interconnectedness and robustness.

*This author's contributions were primarily made while he was affiliated with the University of New Mexico.

Next, we describe an application that measures the *chokepoint potential* of 182 countries—an indicator of how easy it is for a country to control data flowing across its borders.

The main contributions of the paper are: (1) BGPSimPy, which enhances scalability and functionality over earlier projects, is built on open source technologies and is also, itself, open source; (2) simulation results of routing trees for all pairs of ASes, allowing users to compute path information in minutes rather than weeks; (3) illustrations of how BGPSimPy outputs can be used to study network infrastructure; (4) the concept of chokepoint potential as an assessment of how easy it is for a country to control data flowing across its borders.

In the remainder of the paper, we first discuss related work (Section 2), describe BGPSim in Section 3, report results (Section 4), explore chokepoint potential (Section 5), discuss implications and future directions (Section 6) and summarize our contribution in the Conclusion.

2. RELATED WORK

BGP routing tables and Traceroute are widely used resources for inferring Internet topology, and there are several well-known problems that arise using these methods [32, 24, 29, 13]. To address these issues, many researchers have turned to simulation, for example, to simulate BGP at the packet level [11, 30]. These simulators are computationally expensive and impractical for considering all AS pairs. Another approach is to create synthetic AS graphs using tools such as GT-ITM [35], BRITE [25], Inet [21], or ASIM [17] to create graphs that are representative of the Internet. These graphs tend to be much smaller than the actual AS graph, and in some cases do not preserve important graph properties [12, 3].

BGPSim simulates BGP paths, accounting for business relationships that can constrain path choices and using the standard routing policies of local preference, shortest paths, and randomized tie break [13]. BGPSim also encodes export policies, where an AS can export its routes and the routes of its providers and peers to a customer. For a provider or peer, an AS can only export its routes and the routes of its customers. BGPSim creates a unique subgraph for a given AS y, called a routing tree. Paths from any AS x to AS y can then be found using the routing tree for AS y. BGPSimPy builds on the BGPSim tool, using the same heuristics but extending functionality to provide ordered path information and addressing scalability by providing routing trees for all AS nodes. BGPSimPy is built using open-source tools, e.g., Python, Grid Engine [15], and MongoDB, and the code is also open source (`https://github.com/cynthiaw2004/BGPSimPy`).

3. BGPSIMPY

This section gives an overview of BGPSimPy, highlighting similarities and differences from BGPSim.

Following local preference and export policies requires knowledge of AS relationships (peer-to-peer or provider-to-customer). Similar to BGPSim, we use data from Ref. [9] which identifies such relationships. The physical locations of all ASes (by country) in the AS graph were taken from [7]. A path is considered *unreachable* if BGPSimPy does not find a path given a source and destination AS. Paths are discarded if the location of any its constituent ASes cannot be determined from [7].

Internet Exchange Points (IXPs) play an important role in both international Internet routing and Internet censorship. In BGPSimPy, if two ASes are both members of the same IXP, we say that a peer-to-peer edge exists between them, following common practice [5]. We used PeeringDB [28] to determine IXP membership for all ASes considered.

In BGPSimPy, routing trees are saved to disk as sparse adjacency matrices, using an average of 973 KB per matrix. When a path between two ASes is requested, the routing tree for the destination can be reloaded instead of recalculating. 50,979 routing trees totaling 50 GB uncompressed are available for download from `http://www.cs.unm.edu/~freemanc/bgpsimpy/`.

There are three steps to create a routing tree, each involving a breadth-first search on the AS graph. First, a search from the destination node is conducted using only customer-to-provider edges. Single peer-to-peer edges then connect new ASes to the ASes. Finally, new ASes are added to the routing tree using provider-to-customer edges. Ref. [13] reports that all paths can be computed in time $11|V|^2$ with V being the number of ASes. BGPSimPy calculates the routing trees in parallel with each processor allocated a sparse adjacency matrix of the full graph, and then determining a specific routing tree and saving it to disk. Six servers were used to calculate routing trees and paths for all ASes in [9]. Each server was equipped with two Intel Xeon E5-2680 v2 (25M Cache, 2.80 GHz) processors, giving a total of 120 cores and 1.5 TB of RAM.

4. EMPIRICAL RESULTS

This section reports basic statistics on the performance of BGPSimPy and discusses how its output can be used. The following section gives an in-depth example application.

4.1 Performance

On average, it took about 20 minutes to calculate each routing tree. By parallelizing, the wall clock computation time to create all 50,000+ routing trees (one for every AS in our dataset) was about ten days. Table 1 shows the running times for this process broken

out by the three algorithm steps. Step 3 dominates the computation because it fully traverses the partial routing trees produced in steps one and two.

	Overall	Step 1	Step 2	Step 3
min	$2 * 10^{-3}$	$4 * 10^{-4}$	$4 * 10^{-4}$	$6 * 10^{-4}$
max	$2 * 10^{4}$	$8 * 10^{-1}$	$7 * 10^{0}$	$2 * 10^{4}$
std dev	$8 * 10^{2}$	$4 * 10^{-2}$	$3 * 10^{-1}$	$8 * 10^{2}$
mean	$1 * 10^{3}$	$4 * 10^{-2}$	$3 * 10^{-1}$	$1 * 10^{3}$

Table 1: Computing routing trees: Times are reported in seconds for the entire computation and each step.

Once we have the routing tree, computing paths is efficient, taking a mean wall clock time of 0.0086 seconds per path. The minimum time for our dataset was 0.0003 seconds and the maximum was 7.5307 seconds per path.

4.2 Routing Tree and Path Statistics

The data produced by BGPSimPy allow us to ask several questions about the AS graph, e.g., How deep are the trees? Less depth means that the derived paths are likely to be short as well, which is one indicator of AS connectedness [27]. Routing trees can also help assess network robustness. If an AS is disconnected from the network, are there alternative paths to a destination? Are the alternates longer? Although such questions are beyond the scope of this paper, we give some preliminary data here.

Figure 1 shows the frequency of different routing tree depths. The mean path depth is 8.29 and the distribution is positively skewed with a SciPy Stats skew value of 1.19, compared to 0.0 for a normal distribution; there is greater weight on the left tail of the distribution.

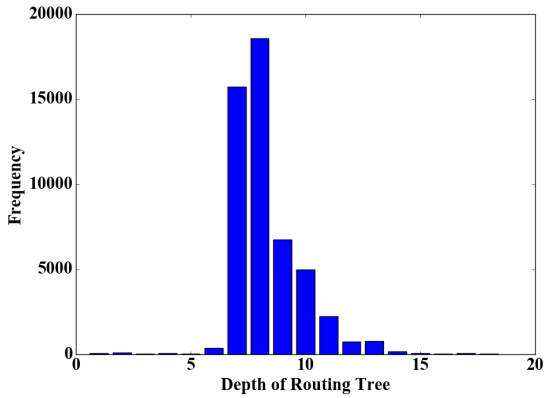

Figure 1: Routing tree depth: The minimum depth is 1, the maximum is 18, and the mean is 8.2904.

The depth 18 ASes include ASNs 29742 and 49487, which are owned by Deluxe Digital Studios in the US and Ztelas in the United Kingdom, respectively.

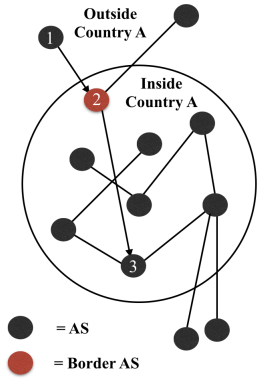

Figure 2: Country A's border is depicted as a circle, and AS 2 is a border node.

We next consider path length. For every AS y, we determined the mean length of all paths that had y as its destination and rounded up the means to the nearest integer. Because the skew value in Figure 1 is positive, most routing trees tend to be short, which suggests that most path lengths will be short as well. Indeed, mean path length frequencies have a positive skew value of 0.2605.

Most paths are of length 4 (data not shown), a result that we validated by comparing to the values reported in earlier work for IPV4 [27] (our data are only for IPV4 as well). That work collected data from route collectors in the Routing Information System and found that path lengths were typically just above 4 hops and than 5.

5. CHOKEPOINT POTENTIAL

In this section, we show how the data provided by BGPSimPy can be used to compare different countries in terms of their ability to control the flow of information across their borders. First, we define *chokepoint potential* as a way to quantify this ability. Chokepoint potential is defined in terms of *border autonomous systems*, or *border nodes*. A border node for country A is an AS inside A that another AS outside of A can reach in one hop (see Figure 2). A country has high chokepoint potential if the majority of AS-level paths in or out of the country pass through a small number border ASes. To find border ASes requires knowing all the ordered paths that enter or exit a country.

We consider chokepoint potential both for *out-to-in* (oti) and *in-to-out* (ito) paths. "Out" and "in" are relative to the country being considered. A path that begins at an AS in China and ends at an AS inside the US is considered an out-to-in path for the US and an in-to-out path for China.

Let $P_{oti}(c)$ represent the set of out-to-in paths with respect to country c. Given a path p, the function $\mathcal{B}^{c}_{oti}$

determines the set of bordernodes in the out-to-in path p with respect to country c. If a path contains a border node b, we consider the path *controllable* by b. The out-to-in chokepoint potential of a bordernode b in country c ($CP_{oti}^{b \in c}$) is determined by:

$$CP_{oti}^{b \in c} = \frac{|\{p \in P_{oti}(c) \text{ s.t. } \mathcal{B}_{oti}^{c}(p) = \{b\}\}|}{\sum_{p \in P_{oti}(c)} |\mathcal{B}_{oti}^{c}(p)|} \quad (1)$$

Since some paths may be controlled by multiple border nodes, we count only those paths controlled by one border node and no others. We then normalize this value by dividing by the total number of border nodes in all paths. We can then aggregate values to determine what percentage of paths into (or out of) c can be controlled by a given number of border nodes. In-to-out chokepoint potential is computed similarly.

We calculated the chokepoint potential for 182 countries. This required knowing the order of ASes in all of the relevant paths, which was not possible previously. The US required the most computation time because it has the most internal ASes (15,072 out of the 50,979 total, see Table 2). Using the MapReduce framework it took 17 hours to determine the denominator (Eq. 1) for in-to-out and 11 hours for the numerator using one VM with 8 x 2.80GHz cores and 64 GB of RAM.

Having computed chokepoint potential for every border node in all 182 countries, we could then make the plot shown in Figure 3a) by ranking the border nodes from largest to smallest in terms of their chokepoint values for selected countries. We then repeated the process for in-to-out as shown in Figure 3b).

Ref. [10] reports that country-level paths are often asymmetric, meaning that the forward path from AS a to AS b does not necessarily match the reverse path. However, the in-to-out and out-to-in chokepoint potentials look remarkably similar for the selected countries (compare Figure 3b) and 3a)). As expected, authoritarian countries such as China and Russia require a small number of border nodes to control a high fraction of paths. Surprisingly, the US Internet resembles China and Russia much more than other Western countries. The US may be an outlier, e.g., the shape of its curve may arise from the historical accident of building out its infrastructure early, and then other countries attaching through preferential attachment.

5.1 Chokepoint Potential and Censorship

With the exception of the US mentioned earlier, Figure 3 shows that countries known for censorship, such as China, can control more paths with fewer border nodes (the curves are much steeper). This observation suggested that there might be a relationship between chokepoint potential and degree of censorship. To study this, we used Freedom on the Net (FOTN) scores [19] from Freedom House, a US-based research institute that studies democracy, political freedom, and human rights [2]. Although FOTN does not directly measure censorship levels, it is a reasonable proxy for censorship and the best measure we found.

Country	Internal	Border	FOTN	FPI
GB	1312	878	24	19
DE	1235	701	18	17
US	15015	2008	19	17
AU	892	341	19	21
RU	3821	911	62	81
CN	271	66	88	85

Table 2: The number of internal nodes (ASes physically located in the country) and border nodes for the six countries in Figure 3 with varying FOTN and FPI scores.

FOTN scores quantify the level of Internet freedom in 65 countries. Each country receives a numerical score from 0 (the most free) to 100 (the least free). Figure 4 depicts FOTN scores worldwide and is taken from [19]. The 2015 FOTN scores are available for only 65 countries and even fewer for earlier years. A similar measure is Freedom House's Freedom of the Press Index (FPI) [18], which contains scores for 182 countries. Like the FOTN score, the FPI provides numerical scores from 0 (the most free) to 100 (the least free) assessing the political, legal, and economic factors that affect access to information. Table 2 shows FOTN and FPI scores for the selected countries in Figure 3. These data show that the countries we selected for Figure 3 represent a spectrum of values, both for FOTN and FPI. China and Russia are at one extreme, and countries like Germany and the United Kingdom are at the other.

We studied how chokepoint potential corresponds to FPI and FOTN. To do this, we asked for each country what percentage of its border ASes was required to control 70% of the paths entering the country (out-to-in chokepoint potential) and compared this to the FOTN score of 65 countries. We fit an ordinary least squares (OLS) model to this dataset, and although the data appear dispersed, the fit is statistically significant ($p \lesssim .026$) with coefficient -36.18. We interpret these results as showing that governments of countries whose network structure facilitates controlling data flowing into the country (fewer border nodes) also have higher FOTN scores (less freedom and likely more censorship). This general pattern holds for in-to-out chokepoint potential and when FPI is substituted for FOTN (Figure 5). In all cases, the OLS fit is statistically significant and shows a negative correlation between chokepoint potential and censorship levels. Although this is a statistical result and we cannot infer causation, it is suggestive that Internet architecture within countries

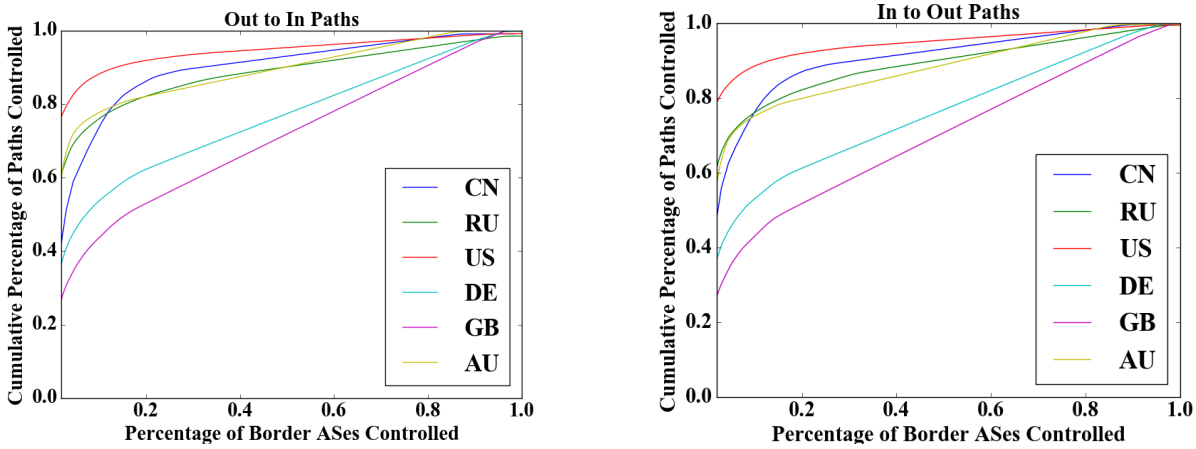

Figure 3: **a)** The percentage of border nodes required to control a given percentage of paths into six different countries. For example, by controlling 20 percent of border ASes, the US government could, in principle, control about 80% of incoming paths. The x-axis ranks the border nodes of a country in descending order of how many paths they control. **b)** The percentage of border nodes required to control a given percentage of paths out of six different countries.

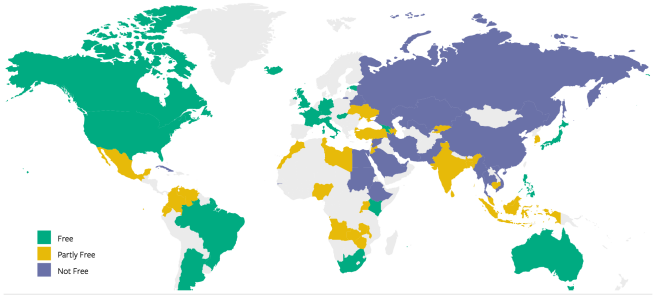

Figure 4: FOTN scores for the year 2015: Colors signify a range of FOTN scores: green (0 - 30), yellow (31 - 60), and purple ($\gtrsim$ 60). Iceland was the most free with a FOTN of 6, and China was the least free with a score of 88.

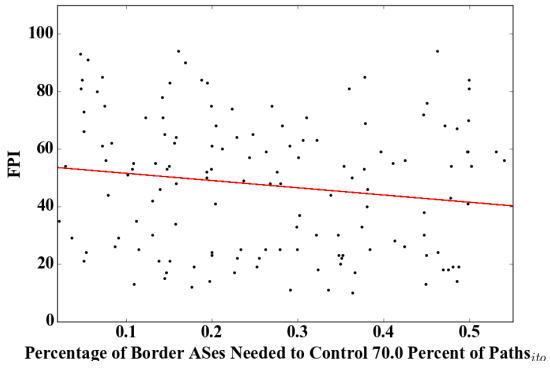

Figure 5: OLS model for the percentage of border ASes required to control 70 percent of outgoing (CP_{ito}) paths vs FPI. The relationship is statistically significant ($p \lesssim .019$) with a coefficient of -25.16.

reflects the tendency of that country to censor cross-border traffic.

6. DISCUSSION

We first highlight the differences between BGPSim and BGPSimPy, discuss some design decisions, and finally give implications and ideas for future work.

6.1 BGPSim vs BGPSimPy

A variety of issues prevented us from conducting a direct performance comparision of BGPSim and BGPSimPy running on identical hardware under identical conditions. For example, the original implementation has access to a number of precomputed paths (see [26]), complicating timing comparisons. Similarly for memory, BGPSim stores routing trees in cache, which are flushed periodically. On a 31.3 GB computer with 15.36 MB of cache, around 30,000 paths could be computed before a flush, whereas in our implementation, all routing trees are computed once and stored on physical disk and are loaded on demand. As mentioned earlier, BGPSimPy outputs the order of ASes for each path computed, which is not available in BGPSim. Section 5 illustrates how that information can be useful.

6.2 Alternative Censorship Measures

Other indicators could be used to estimate censorship besides FOTN and FPI. Polity IV [1] is a dataset that contains annual information on the level of democracy for all states with greater than 500,000 total population. There is a strong statistical correlation of -0.81 between FOTN scores and Polity IV data. We selected FOTN because of its greater relevance to our topic, even

though Polity IV score is measured across a wider range of nations.

The OpenNet Initiative (ONI) [20] has published a summary of global Internet filtering data in 74 countries. Although relevant, these data are coarse-grained and do not produce a single score, while the FOTN scores could be used directly.

6.3 Chokepoint Potential

There are several ways that a country can create chokepoints, either intentionally or accidentally: taking advantage of Internet Exchange Points, limiting the number of physical connections that cross the border, or working with—and supporting—a small set of large ISPs that have international connections. Each of these combines two basic elements: limiting the number of organizations that the government must interact with and creating physical locations where traffic can be controlled. We chose to define chokepoint potential in terms of the AS graph instead of physical locations because the AS graph captures most interesting physical attributes of the Internet, and virtualization of the physical and datalink layers on the Internet make physical locations less meaningful. A single undersea cable can be multiplexed to serve as several different links in the AS graph, and ASes in one country can have physical Points of Presence (PoPs) in several other countries (*e.g.*, China Telecom in Pasadena, California). Thus, the AS graph is a good approximation of overall chokepoint potential, and information such as geography and the physical locations of routers does not add meaningfully to the analysis without considering every individual piece of Internet infrastructure on a case-by-case basis.

6.4 Future Work

One compelling question to be answered in future work is why the US has such high chokepoint potential compared to other countries in Figure 3. One hypothesis is that preferential attachment—that is, the tendency for new ASes to connect to ASes that are already well connected—causes more mature ASes to be well-connected, thus creating chokepoints for purely historical reasons. Alternatively, in some rankings [20] that analyze surveillance in addition to censorship score, the US does not appear among the freest countries.

We are also interested in understanding the causal effects of censorship, which would require temporal measurements of Internet infrastructure and censorship. By taking measurements before and after a country implements new laws on censorship, we could observe if and when a significant change in AS topology occurs. Lists of AS relationships for different years are available at [34]. We are currently running BGPSimPy to determine routing trees for different years and hope to compare changes in the topology, if any.

Ref. [22] discusses the topological measure of *country centrality*, which estimates the impact that each country has on reachability between other countries. This would be an interesting statistic to compare to FOTN scores and FPI.

7. CONCLUSION

Over 3 billion people in the world have access to the Internet, and 34% of them live under governments that disconnect Internet or mobile phone access, often for political reasons [19]. Governments in Sudan [19], Egypt [14], Pakistan [23], Vietnam [19], and Iran[4] have either intentionally disconnected their citizens from the Internet during times of crisis or influenced their network topology to facilitate censorship and other controls. Given these trends, large-scale studies of Internet topology and BGP routes will remain important and require the addtional scalability and functionality provided by tools like BGPSimPy.

This paper describes an implementation that stores routing trees on physical disk, bypassing the memory issues associated with earlier systems, and it provides the exact AS order of computed paths. Over the course of approximately two weeks, routing trees and paths were found for 50,000+ ASes. We illustrated the value of these data in a study of the chokepoint potential of several different countries. We observed that countries that practice more censorship, as measured by FOTN and FPI scores, tend to have higher chokepoint potential. This correlation may reflect the desire of some countries to control data flowing over its borders. Studies like these are enabled by our extension of BGPSim. There are many avenues for future work using BGPSimPy as outlined in Section 6.4.

8. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the partial support of NSF (1518878,1444871), DARPA (FA8750-15-C-0118), AFRL (FA8750-15-2-0075), the Sandia National Laboratories Academic Alliance, and the Santa Fe Institute.

This material is based upon work supported partially by the NSF (1314297, 1420716, 1444871, 1518523, and 1518878); DARPA (FA8750-15-C-0118); and AFRL FA8750-15-2-0075). Ian Beaver, Cari Martinez and Padraic Cashin provided valuable technical assistance. Phillipa Gill, Rachee Singh, and Rishab Nithyanand generously helped us with installation and early experiments using BGPSim. NextIT provided the computational resources for our experiments.

9. REFERENCES

[1] Polity IV annual time-series, 1800,2014. http://www.systemicpeace.org/polity/polity4.htm.

[2] Cuba after fidel - what next? *Voice of America*, 2009.

[3] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore. On the bias of traceroute sampling: or, power-law degree distributions in regular graphs. *Journal of the ACM (JACM)*, 56(4):21, 2009.

[4] C. Anderson. Dimming the internet: Detecting throttling as a mechanism of censorship in iran. *arXiv preprint arXiv:1306.4361*, 2013.

[5] B. Augustin, B. Krishnamurthy, and W. Willinger. Ixps: mapped? In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 336–349. ACM, 2009.

[6] I. Avramopoulos, M. Suchara, and J. Rexford. How small groups can secure interdomain routing. Technical report, Technical report, Princeton University Comp. Sci, 2007.

[7] T. Bates, P. Smith, and G. Huston. The cidr report, 2011.

[8] H. Chan, D. Dash, A. Perrig, and H. Zhang. Modeling adoptability of secure bgp protocol. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 279–290. ACM, 2006.

[9] List of as relationships from cyclops. https://raw.githubusercontent.com/sbunrg/Astoria/master/bgp_sim/Cyclops_caida_cons.txt.

[10] A. Edmundson, R. Ensafi, N. Feamster, and J. Rexford. Characterizing and avoiding routing detours through surveillance states. *arXiv preprint arXiv:1605.07685*, 2016.

[11] N. Feamster, J. Winick, and J. Rexford. A model of bgp routing for network engineering. In *ACM SIGMETRICS Performance Evaluation Review*, volume 32, pages 331–342. ACM, 2004.

[12] P. Gill, M. Schapira, and S. Goldberg. Let the market drive deployment: A strategy for transitioning to bgp security. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 14–25. ACM, 2011.

[13] P. Gill, M. Schapira, and S. Goldberg. Modeling on quicksand: Dealing with the scarcity of ground truth in interdomain routing data. *ACM SIGCOMM Computer Communication Review*, 42(1):40–46, 2012.

[14] J. Glanz and J. Markoff. Egypt leaders found off switch for internet. *The New York Times*, 2011.

[15] http://gridscheduler.sourceforge.net/.

[16] T. G. Griffin and G. Wilfong. An analysis of bgp convergence properties. *ACM SIGCOMM Computer Communication Review*, 29(4):277–288, 1999.

[17] P. Holme, J. Karlin, and S. Forrest. An integrated model of traffic, geography and economy in the internet. *ACM SIGCOMM Computer Communication Review*, 38(3):7–15, 2008.

[18] F. House. Freedom of the press index. *Freedom House*, 2011.

[19] F. House. Freedom on the net 2015, privatizing censorship, eroding privacy, 2015.

[20] O. Initiative. Summarized global internet filtering data spreadsheet, 2012.

[21] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. 2000.

[22] J. Karlin, S. Forrest, and J. Rexford. Nation-state routing: Censorship, wiretapping, and BGP. *arXiv preprint arXiv:0903.3218*, 2009.

[23] S. Khattak, M. Javed, S. A. Khayam, Z. A. Uzmi, and V. Paxson. A look at the consequences of internet censorship through an ISP lens. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 271–284. ACM, 2014.

[24] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate as-level traceroute tool. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378. ACM, 2003.

[25] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pages 346–353. IEEE, 2001.

[26] https://github.com/sbunrg/Astoria/tree/master/bgp_sim.

[27] https://labs.ripe.net/Members/mirjam/interesting-graph-as-path-lengths.

[28] Peeringdb. https://www.peeringdb.com/.

[29] J. Qiu and L. Gao. As path inference by exploiting known as paths. In *Proceedings of IEEE GLOBECOM*. Citeseer, 2005.

[30] B. Quoitin and S. Uhlig. Modeling the routing of an autonomous system with c-bgp. *IEEE network*, 19(6):12–19, 2005.

[31] R. Singh, H. Koo, N. Miramirkhani, F. Mirhaj, P. Gill, and L. Akoglu. The politics of routing: Investigating the relationship between as connectivity and internet freedom. In *6th USENIX Workshop on Free and Open Communications on the Internet (FOCI 16)*, Austin, TX, 2016. USENIX Association.

[32] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the*

IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, pages 618–627. IEEE, 2002.

[33] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin. Internet routing resilience to failures: analysis and implications. In *Proceedings of the 2007 ACM CoNEXT conference*, page 25. ACM, 2007.

[34] `http://irl.cs.ucla.edu/topology/ipv4/relationship/`.

[35] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, volume 2, pages 594–602. IEEE, 1996.