

Harnessing Internet Topological Stability in Thorup-Zwick Compact Routing

Stephen D. Strowes
School of Computing Science
University of Glasgow
Email: sds@dcs.gla.ac.uk

Colin Perkins
School of Computing Science
University of Glasgow
Email: csp@csperkins.org

Abstract—Thorup-Zwick (TZ) compact routing guarantees sublinear state growth with the size of the network by routing via landmarks and incurring some path stretch. It uses a pseudo-random landmark selection designed for static graphs, and unsuitable for Internet routing. We propose a landmark selection algorithm for the Internet AS graph that uses k -shells decomposition to choose landmarks. Using snapshots of the AS graph from 1997–2010, we demonstrate that the ASes in the k_{max} -shell are highly-stable over time, and form a sufficient landmark set for TZ routing in the overwhelming majority of cases (in the remainder, adding the next k -shell suffices). We evaluate path stretch and forwarding table sizes, and show that these landmark sets retain low average path stretch with tiny forwarding tables, but are better suited to the dynamic nature of the AS graph than the original TZ landmark selection algorithm.

I. INTRODUCTION

Sustainable inter-domain Internet routing requires a scalable forwarding plane, but shortest path routing algorithms for general graphs have forwarding table size $O(n)$, where n is the number of nodes. This affects memory requirements in routers, and may lead to scaling problems as the network grows [1], [2], enforcing continuous router upgrades for many network operators. Compact routing algorithms, on the other hand, offer provable worst-case sublinear forwarding state growth, by allowing path stretch within a well-defined bound.

Thorup-Zwick (TZ) compact routing for weighted, undirected, static graphs guarantees worst case multiplicative path stretch 3, with forwarding table size $O(\sqrt{n \log n})$ [3] (performance on AS graph snapshots is considerably better than this theoretical bound [4]). At each node, the algorithm derives a forwarding table containing a globally visible set of landmark nodes A , and a subset of the rest of the graph. Forwarding uses landmarks only if the destination is locally unknown.

TZ landmark selection is centralised and designed for static graphs. It iteratively grows the landmark set from an initial random set of nodes. However, in a dynamic graph, a stable set of landmarks is required in the presence of change; this must be computable in a distributed manner. To demonstrate a feasible set of stable landmarks for the Internet, we use the k -shells graph decomposition of daily AS graph snapshots derived from BGP data collected between Nov 1997 and Nov 2010 [5]. The k -shells algorithm on AS graphs reveals a well-connected *nucleus* corresponding to the k_{max} shell (see [6] and §III). We show the nucleus is remarkably stable over time.

We define a new landmark selection algorithm using k -shells decomposition to form landmark sets suitable for TZ compact routing. We show that, in the overwhelming majority of cases, the nucleus alone satisfies TZ’s requirements for sub-linear forwarding table growth. In the remaining 1.8% of cases, one additional k -shell is needed (§V-A). Advertisement of changes to this set would incur low overhead in a decentralised TZ routing protocol. We present path stretch and forwarding table size results for TZ_k landmark sets on the AS graph, and show that they are comparable to standard TZ landmark sets.

Our contributions are as follows. We present an in-depth study of the stability of the AS graph nucleus as determined by the k -shells graph decomposition. Next, we describe TZ_k , a modification to TZ compact routing that uses the k -shells decomposition to provide a stable and long-lived landmark set. No compact routing algorithm has previously used k -shells decomposition to intelligently place landmarks. Finally, we demonstrate that TZ_k performs well on real-world graphs, matching TZ performance.

II. RELATED WORK

Inter-domain routing scalability with shortest path routing is an ongoing concern in the IPv4 Internet [1], [2] as forwarding tables grow. Longer-term, an issue for router design is forwarding table lookup times, and the associated costs for adding or removing prefixes from tables [7]. IPv6 has the same scaling issues as IPv4, with potentially more prefixes.

To achieve sublinear state growth at all nodes in a general network, we must accept worst-case path stretch of 3 [8].

The AS graph node degree distribution follows a power law [9]. The compact routing algorithms of Brady and Cowen [10] and Chen *et al.* [11] use this to improve performance. Chen *et al.* use node degrees to select landmarks, reducing forwarding state compared to the theoretical bound in general graphs. The Brady-Cowen (BC) algorithm constructs a spanning tree rooted at the highest-degree node, then additional smaller trees on connected regions at the edge of the network, some number of hops away from the root of the primary tree; nodes use distance labelling to determine the spanning tree with fewest hops to the destination. The TZ and BC algorithms perform well on synthetic power-law random graphs [12]–[14] and AS graph snapshots [4], with mean stretch ~ 1.1 . Path stretch with TZ routing (§IV-A) is more consistent than BC routing [4].

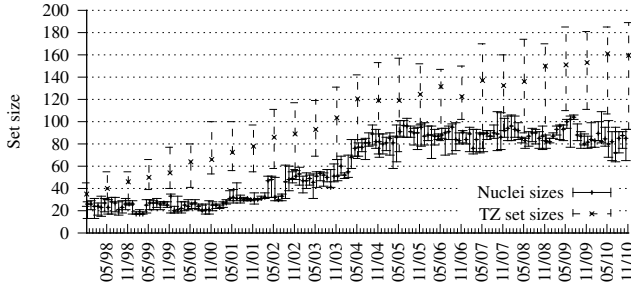


Fig. 1. k -shells nucleus growth and TZ landmark set growth for snapshots of the AS graph, showing the 5th percentile, median, and 95th percentile.

III. STABILITY OF k -SHELLS DECOMPOSITION

From [15], a subgraph $G' = (V', E|V')$ induced by the set $V' \subseteq V$ is a k -shell iff $\forall v \in V' : \deg_{G'}(v) \geq k$ and G' is the maximum subgraph with this property. A node v has shell index k if it belongs to the k -shell, but not the $(k + 1)$ -shell. A k -shell, then, is all nodes whose shell index is k [16]. All k -shells of a graph can be obtained by recursively removing all nodes of degree $< k$ until all nodes in the remaining graph have degree $\geq k$, incrementing k when no additional nodes qualify for that k -shell. The k -shells decomposition exposes structure in a graph that is not obvious from node degree alone.

The largest k that generates a non-empty k -shell is k_{max} . The k_{max} -shell is the *nucleus*: a highly-connected component in the core of the network. Using the nucleus for landmarks ensures that they are structurally important in the network.

We use AS graphs to show desirable properties of a k -shell based landmark set for TZ routing on the Internet. To build AS graphs dating Nov 1997–Nov 2010, we collect the unique set of BGP paths from all Route Views collectors available each day, determine the set of links between ASes, and generate k -shells decompositions of these daily snapshots.

The graph has grown from 3030 ASes in Nov 1997 to 36255 ASes in Nov 2010. We study the nucleus for the same period. Frequent nucleus size changes would suggest the network is not stable enough to support TZ-based routing, while a slowly varying nucleus suggests stability to be exploited. Fig. 1 shows growth of the nucleus, aggregating the range of sizes each month and plotting median, 5th, and 95th percentiles. While growth is not linear, as might be expected from growth of the network [17], we show in §V-A that these are valid landmark sets in the vast majority of cases. We posit that the flattening growth curve is a sampling artefact: many Route Views collectors were added from 1997–2005, with few since. AS graphs derived from BGP data are highly accurate near collectors, but links toward edge networks may not be visible [18]. This effect will grow unless the number of collectors scales with the growth of the network, leading to the curve in Fig. 1.

The variation in landmark set size with the standard TZ algorithm is also shown in Fig. 1. These sets are larger than the nucleus, but §V-A shows that the nucleus is sufficient as a landmark set in the majority of cases. By extension, any larger nucleus revealed in a fuller graph will therefore also suffice.

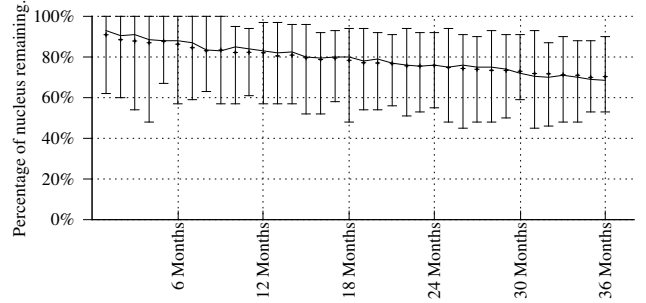


Fig. 2. Remainder rate. Bars show minima/maxima; points are medians; the line tracks the mean.

In an evolving network, the landmark set will change over time, but a generally stable landmark set is desirable to reduce the overhead of recomputing and readvertising forwarding tables. We measure departures from the landmark set by calculating the fraction of the set A_{t_1} from time t_1 that is still present in the landmark set A_{t_2} at time t_2 . Fig. 2 plots this at fixed intervals up to 3 years from the initial date. We choose 3 years as a long-term measure in terms of Internet evolution. The departure rate is clearly linear, with 70% (mean and median) of a nucleus remaining after the 3 years. Given the rate of growth of the network, one might reasonably expect greater instability; to observe such stability is a key result. We find that the lower bound of the range in Fig. 2 is introduced primarily by the networks earlier in our dataset with smaller nucleus sets, and the upper bounds of the range is introduced by more recent networks.

These results show a long-term stability at the centre of the network. Fig. 3 shows when each of the 245 ASes that have appeared in the nucleus over our dataset were present. This demonstrates at a high-level the stability of the set, with a tendency toward older ASes (those with lower AS numbers).

Fig. 4 shows the fraction of the network directly connected to the nucleus for transit services, derived from inferred AS relationships data [19]. Due to network growth, we see a small reduction in the fraction of the network directly connected to the nucleus, though this may be due to the reduced accuracy noted earlier. Almost 50% of the network is directly connected to this relatively small set of nodes. The mean AS hop count from the nucleus to all other nodes is in the range 2.5 – 2.7; the 99th percentile is 4 or 5 hops. These places the nucleus at the heart of the network: the mean distance between any two pairs of nodes in the AS graph has consistently been between 3.5 – 4.0 during these dates; the maximal diameter has risen from 9 hops to 11. On this basis, the nucleus appears to be located as the network’s natural core.

It is important that the landmark set be geographically distributed, so latency from additional hops is small. The Nov 2010 nucleus contains 100 ASes. We infer geographic placement of these using WHOIS data: 45 are in North America; 42 in Europe; 12 in Asia-Pacific; 1 in Africa. They span 27 different countries. LACNIC region is not represented, though this would change as demand improves network deployment.

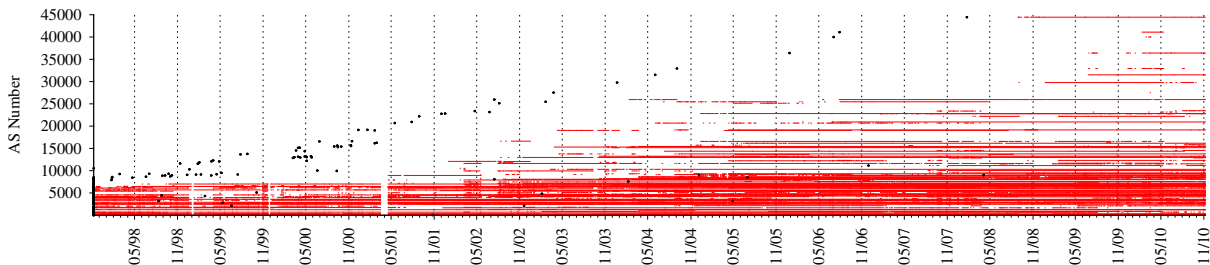


Fig. 3. Dates ASes appear in the nucleus. Black dots show when an AS was first seen; horizontal bars show dates it appears in the nucleus. More than 45,000 unique ASes have been seen; 245 appear in the nucleus. Of these, 212 were still in the BGP data, even if not in the nucleus, on 8 Nov 2010.

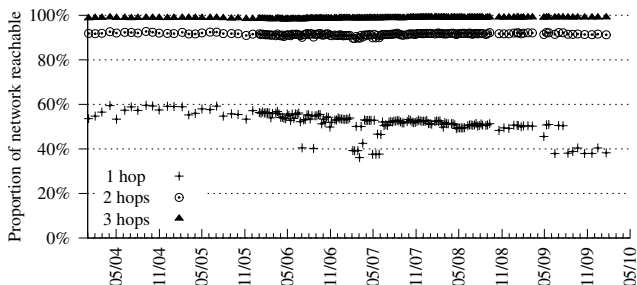


Fig. 4. Percentage of customers/peers n -hops from nuclei.

IV. TZ COMPACT ROUTING WITH k -SHELLS LANDMARKS

Using k -shells decomposition as the basis for the landmark set exposes a highly connected, highly visible region of the network. This region is structurally important: 30% of nodes are reachable only via the nucleus, and distances between the remainder are shortened considerably by routing through the nucleus [6]. Choosing the nucleus as a landmark set has no stretch for many paths, and low stretch for the others (§V-B).

A. TZ Compact Routing and Landmark Selection

Consider a graph $G = (V, E)$, where V is the set of all nodes and E is the set of all edges. TZ compact routing defines a set of landmark nodes $A \subseteq V$ towards which every node $v \in V$ retains paths. Each node v is surrounded by a cluster C_v containing every node w that is fewer hops from v than from the nearest node in A . Forwarding table entries are maintained for nodes in C_v . Given V , a function $d(v, w)$ that computes the minimum distance between nodes v and w , and function $D(v, A) = \min_{w \in A} \{d(v, w)\}$, at each node v the cluster set C_v is:

$$C_v = \{w \in V \mid d(v, w) < D(w, A)\} \quad (1)$$

The TZ landmark selection algorithm produces an expected landmark set size of $O(s \log n)$, where $s = \sqrt{n / \log n}$ and $n = |V|$ [3]. It is an iterative algorithm, which for each iteration i grows the landmark set A to be the union of itself and a subset of nodes chosen independently from a set W_i with probability $s/|W_i|$. On the first iteration, $W_0 = V$, and so A is therefore populated with a random selection from the full

graph. In each subsequent iteration i , large clusters are broken up by repopulating W_i according to the following equation:

$$W_i \leftarrow \{v \in V \mid |C_v| > 4n/s\} \quad (2)$$

The algorithm terminates on the first iteration that $W_i = \emptyset$. The cluster size constraint in Eq. 2 exists to maintain bounded forwarding table sizes.

The TZ landmark set selection algorithm requires structured labels, rather than simple addresses, to route packets. A destination's label comprises its name, the name of one of its nearest landmarks, and the next hop from the landmark to allow it to forward packets into the correct cluster. Paths do not necessarily include the landmark: a path that uses the landmark is the *longest* possible TZ path between two nodes. It is possible, and expected, that packets bound for destination d with landmark l will arrive at an intermediate node which has, through the clustering process, retained a reference to d , and thus skip l altogether and short-cut toward d .

In Fig. 1 we compare representative landmark set sizes for the TZ compact routing algorithm with the size of the nucleus for each snapshot. We generated fifty TZ landmark sets for the same dates as evaluated in §V, and plot the range of landmark set sizes (using the median value with 5th and 95th percentiles). Landmark sets generated by the TZ landmark selection algorithm are seen to vary considerably in size, due to pseudo-random landmark selection, offering little stability.

Algorithms that rely on node degree as a measure of importance [11] are unlikely to be appropriate for the Internet. High degree may indicate a node with many customers and few providers, or one with many providers and a high level of peering with other transit nodes. It is more important that landmarks are well-connected to the wider network and long-lived (hence less likely to fail). The k -shell decomposition of a graph gives a landmark set with such properties (§III).

B. TZ Routing with k -shells: TZ_k

We combine TZ compact routing with k -shells graph decomposition to produce a modification to the TZ routing algorithm which replaces the landmark selection algorithm described in §IV-A. We refer to the result as TZ_k . As our algorithm uses the structure of the network and eliminates the random element of TZ, our landmark sets are deterministic.

Algorithm 1 : landmark(G, s)

Generate k -shells $k = 1, 2, \dots, \max - 1, \max$
 $A \leftarrow \emptyset$; $W = \emptyset$; $i \leftarrow \max$
do
 $A \leftarrow A \cup i$ -shell
 $i \leftarrow i - 1$
 $C_v = \{w \in V \mid d(v, w) < D(w, A), \text{ for every } v \in V\}$
 $W = \{v \in V \mid |C_v| > 4n/s\}$
while $W \neq \emptyset$
return A

Our new landmark selection algorithm is shown in Alg. 1. Given a static graph, and setting the parameter s as defined in §IV-A, we first determine the k -shells for the graph. Then, starting from the nucleus, the k_{\max} -shell, we test the clustering constraints (Eq. 2). If the test fails, we expand the landmark set to include the $(k_{\max} - 1)$ -shell, and so forth. Building landmark sets in this manner no longer *guarantees* that we meet the TZ compact routing constraints, but they are met on *all* Internet snapshots studied, and more generally should be met on other power-law graphs. This algorithm selects topologically well placed ASes as landmarks. §V shows that it provides *at least* the same performance as TZ, but with the benefit that it will select as landmarks a highly-stable, geographically distributed set of ASes that are likely already performing transit functions.

V. PERFORMANCE OF TZ_k ON THE AS GRAPH

A. Constraint Checking

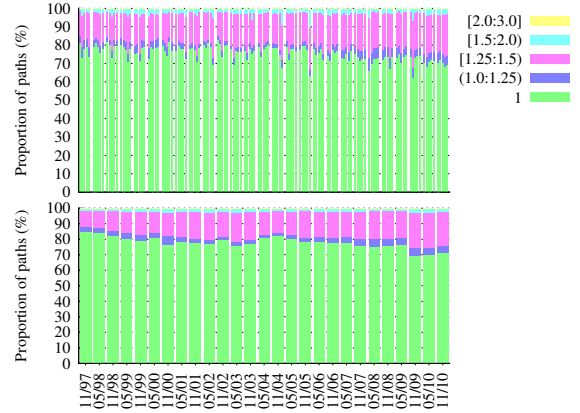
Alg. 1 determines the nucleus to be a sufficient and valid landmark set in 98.2% of the 4662 snapshots (i.e., on the first iteration, the algorithm does not produce any clusters violating Eq. 2). Only 85 (1.8%) snapshots require a second iteration; generally only one node had a cluster breaking the size constraint. A third iteration was never required.

Despite massive growth, the network remains extremely well-suited to the landmark sets generated by TZ_k, and the k_{\max} -shell remains centrally located and well-connected for use as a landmark set for TZ routing.

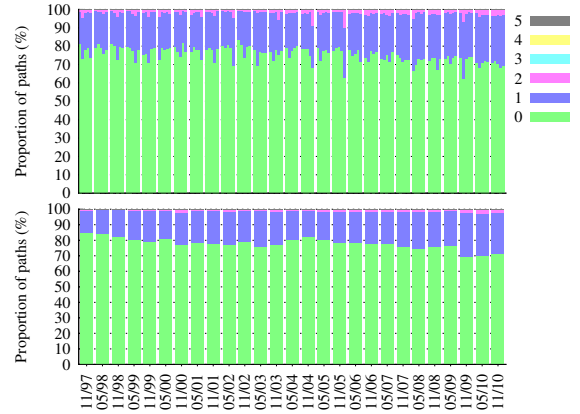
B. Path Stretch

Taking one snapshot every six months across the full dataset, we use Alg. 1 to generate landmark sets. Using these, we construct forwarding tables as defined by [3], with the addition of neighbour nodes into all forwarding tables following [12]. To analyse stretch, we simulate packet forwarding from each node to 1% of the other nodes, chosen with uniform probability. We also simulate forwarding along the reverse path.

Fig. 5(a) shows multiplicative stretches observed with TZ and TZ_k respectively (for TZ, from fifty landmark sets generated for each snapshot we show a subset of five experiments spanning the range of landmark set sizes). We observe that 75.1% of all TZ_k paths tested are stretch 1, while 90.0% of all paths tested have stretch < 1.3; only 0.09% of paths tested are stretch ≥ 2.0 . The maximum observed value for the range [2.0 : 3.0] on all of the graphs we tested is 0.27%, in



(a) Multiplicative stretches for TZ (top) and TZ_k (bottom).



(b) Additive stretches for TZ (top) and TZ_k (bottom).

Fig. 5. Path stretch for the TZ and TZ_k algorithms.

Nov 2009. Our TZ_k stretch results are comparable to our TZ stretch results and previous work [4], [14]. Network growth has introduced a gradual reduction in the number of paths experiencing any stretch, but the results indicate that TZ_k continues to perform extremely well on these networks, with the majority of paths experiencing no stretch.

Additive stretch results, Fig. 5(b), tell how many extra hops were used. 21.3% of all paths are stretched by only 1 hop. Only 1.3% of paths experience more than one extra hop, with the proportion diminishing rapidly as the number of extra hops increases. Again, our TZ_k results are comparable with TZ.

As clusters can overlap and are not symmetric (given nodes a and b , C_a containing b does not imply C_b contains a), it is possible for path lengths to differ and path stretch to be non-symmetric. We find that 64.7% of paths are symmetric; 86% of those are stretch 1 in both directions. Of the 35.3% asymmetric paths, 99.8% were stretch 1 in one direction. Of all paths observed, 82.5% showed stretch < 1.3 in both directions. No paths were stretch 3 in both directions

Stretch performance with our landmark sets is consistently good. Long stretch is rare; paths are predominantly shortest-path. Although AS graphs derived from BGP data can be

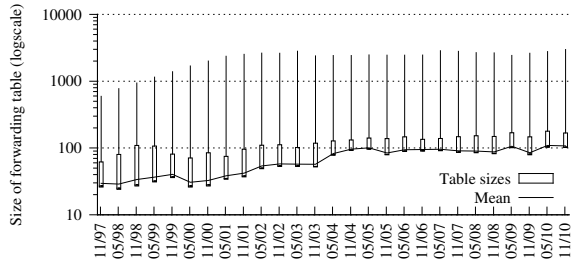


Fig. 6. TZ_k forwarding table sizes. Boxes show minima, 1st percentile, median, 99th percentile, and maximum; minima are not visible.

incomplete [18], [20], we believe our stretch results are worst-case since additional usable paths will improve performance. An operational network might manually insert routes for popular destinations to ensure low-stretch paths, although this would increase forwarding table sizes.

C. Forwarding Table Sizes

We show the distribution of forwarding table sizes using TZ_k landmarks on AS graphs in Fig. VI. Even though we additionally insert all neighbours into forwarding tables, we find that those tables are extremely small, generally a few tens of entries. By adding all neighbours, the largest tables are two orders of magnitude larger than plain TZ due to the degree distribution of the nodes in the network, but this is small compared to storing references for all ASes, as in BGP. On the Nov. 2010 snapshot, only 5 nodes maintain forwarding entries for >5% of the network, and none reference more than 9% of the network. 99.8% of nodes retain forwarding entries for <1% of the full network. Other snapshots show very similar forwarding table size distributions. Forwarding tables for TZ_k have the desirable property that they are consistently small.

VI. FUTURE WORK

Distributed Computation of k -Shells: For real-world use, the TZ_k algorithm must permit distributed computation of the landmark set. A distributed algorithm for computing a k -shell, given a parameter k , is outlined in [21]. However, our goal is to find the k_{max} -shell, so we must modify the algorithm. Once recomputed, changes to the landmark set must be announced, resulting in only partial recomputation of routing state at some other ASes. For a future Internet protocol, a decentralised k -shells algorithm requires ASes to share information about the ASes they have agreements with. Much of this is exposed via BGP in the current network.

State Requirements: Each AS that is not a landmark has at least one other AS that acts as its landmark. If an AS, d , is equidistant from many landmarks, then any of them is on a valid path to d . A mapping system that reveals the set of valid landmarks for d allows a source s to select the nearest. This minimises the distance $s \rightarrow L_d$, and as all $L_d \rightarrow d$ are the same length, a greater proportion of shortest-paths are used.

For local forwarding state, a node v retains all nodes closer to v than to their own landmarks (Eq. 1). Thus cluster compu-

tation isn't as simple as defining a low TTL for routing updates from v . Routing updates must contain the distance between the origin and its nearest landmarks, to provide other ASes with the information required to perform the clustering calculation. The additional information allows scoped propagation.

Policy: Routing policy, managed in BGP via path inflation and modification of local preferences, is not considered by any compact routing algorithm. No architecture that routes on AS number alone can offer fine-grained control over prefixes.

VII. CONCLUSIONS

We present an analysis of the k -shells decomposition of the AS graph using data spanning 14 years, and show that it has remained stable despite the network's growth. The nucleus of this set offers a highly-visible, well-connected, stable, and long-lived set of nodes that is topologically well-placed. We defined TZ_k , a variant of the TZ compact routing algorithm that adopts this nucleus as the basis of its landmark set. It offers the same consistently excellent routing performance on AS graphs as the TZ routing algorithm, achieving shortest-path routes in the majority of cases, and generating extremely small forwarding tables. The mean path stretch is small (~ 1.1) and usually adds only one additional AS hop. Adoption of a stable landmark set in TZ_k is a necessary step towards deployment of a decentralised routing protocol based on the TZ algorithm, and the application of compact routing to the Internet.

REFERENCES

- [1] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," RFC 4984, September 2007.
- [2] T. Li, "Recommendation for a Routing Architecture," RFC 6115, 2011.
- [3] M. Thorup and U. Zwick, "Compact routing schemes," in *SPAA*, 2001.
- [4] S. D. Strowes, G. Mooney, and C. S. Perkins, "Compact Routing on the Internet AS-Graph," in *Global Internet Symposium*, April 2011.
- [5] Route Views Project, <http://www.routeviews.org/>, University of Oregon.
- [6] S. Carmi *et al.*, "A model of Internet topology using k -shell decomposition," *Proc. Nat. Acad. Sciences*, vol. 104, no. 27, 2007.
- [7] K. Fall, G. Iannaccone, S. Ratnasamy, and P. B. Godfrey, "Routing tables: Is smaller really much better?" in *HotNets*, October 2009.
- [8] L. J. Cowen, "Compact routing with minimum stretch," in *Proc. 10th ACM/SIAM Symposium on Discrete Algorithms*, January 1999.
- [9] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law Relationships in the Internet Topology," in *SIGCOMM*, 1999, pp. 251–262.
- [10] A. Brady and L. Cowen, "Compact routing on power law graphs with additive stretch," in *ALLENEX*, 2006.
- [11] W. Chen *et al.*, "Compact routing in power-law graphs," in *Proceedings of the 23rd International Symposium on Distributed Computing*, 2009.
- [12] D. V. Krioukov and K. C. Claffy, "Toward compact interdomain routing," *CoRR*, vol. abs/cs/0508021, 2005.
- [13] D. Krioukov, K. Claffy, K. Fall, and A. Brady, "On Compact Routing for the Internet," in *SIGCOMM CCR*, vol. 37, no. 3, July 2007, pp. 41–52.
- [14] D. Krioukov, K. Fall, and X. Yang, "Compact Routing on Internet-Like Graphs," in *Infocom*, March 2004.
- [15] V. Batagelj and M. Zaversnik, "Generalized Cores," *CoRR*, vol. cs.DS/0202039, 2002.
- [16] J. I. Alvarez-Hamelin *et al.*, " k -Core Decomposition of Internet Graphs," *Networks and Heterogeneous Media*, June 2008.
- [17] B. E. Carpenter, "Observed Relationships between Size Measures of the Internet," *SIGCOMM CCR*, vol. 39, no. 2, pp. 5–12, 2009.
- [18] R. Oliveira *et al.*, "The (in)Completeness of the observed Internet AS-level structure," *IEEE/ACM Trans. Networking*, vol. 18, no. 1, Feb. 2010.
- [19] AS Relationships, <http://www.caida.org/data/active/as-relationships/>.
- [20] M. Roughan *et al.*, "Bigfoot, Sasquatch, the Yeti and Other Missing Links: What We Don't Know About the AS Graph," in *IMC*, 2008.
- [21] S. Gangam and S. Fahmy, "Distributed Partial Inference under Churn," in *Global Internet Symposium*, March 2010.