Near-Deterministic Inference of AS Relationships

Yuval Shavitt
School of Electrical Engineering
Tel-Aviv University, Israel
Email: shavitt@eng.tau.ac.il

Eran Shir
School of Electrical Engineering
Tel-Aviv University, Israel
Email: shire@eng.tau.ac.il

Udi Weinsberg
School of Electrical Engineering
Tel-Aviv University, Israel
Email: udiw@eng.tau.ac.il

***Abstract*—The discovery of Autonomous Systems (ASes) interconnections and the inference of their commercial Type of Relationships (ToR) has been motivated by the need to accurately calculate AS-level paths. An inherent problem in current algorithms is their extensive use of heuristics, causing unbounded errors that are spread over all inferred relationships. We propose a near-deterministic algorithm for solving the ToR inference problem that uses the Internet's core, a dense sub-graph of top-level ASes. We test several methods for creating such a core and demonstrate the robustness of the algorithm to the core's size and density, the inference period, and errors in the core.**

We evaluate the algorithm using AS-level paths collected from RouteViews BGP paths and DIMES traceroute measurements. Our proposed algorithm deterministically infers over 95% of the approximately 58,000 AS topology links using a week worth of data and as little as 20 ASes in the core. The algorithm infers 2–3 times more peer-to-peer relationships in links discovered only by DIMES than in RouteViews, validating the need for a broad and diverse Internet measurement effort.

I. Introduction

Today's Internet consists of thousands of autonomously administrated networks (ASes), each is assigned with one or more blocks of IP prefixes. ASes communicate routing information to each other using the Border Gateway Protocol (BGP), and use a set of local policies for selecting the best route for each reachable prefix. The propagation of selected routes to neighboring ASes is subject to local policies and rules, that are largely restricted by the Type-of-Relationship (ToR) between ASes. In order to calculate the feasible paths between ASes, one needs to obtain the ToR between all neighboring ASes. Since ToRs are regarded as proprietary information, deducing them is an important yet difficult problem.

There are three major commercial relationships between neighboring ASes [15]: customer-to-provider (c2p), peer-to-peer (p2p), and sibling-to-sibling (s2s). In the c2p category, a customer AS pays a provider AS (usually larger than the customer) for traffic that is sent between the two. In the p2p category, two ASes freely exchange traffic between themselves and their customers, but do not exchange traffic from or to their providers or other peers. In s2s, two ASes administratively belong to the same organization and freely exchange traffic between their providers, customers, peers, or other siblings.

Gao [9] was the first to study the AS relationships inference problem and deduced that every BGP path must comply with the following pattern: an uphill segment of zero or more c2p or s2s links, followed by zero or one p2p links, followed by a downhill segment of zero or more p2c or s2s links. Paths with this hierarchical structure are called *valley-free* or valid. Paths that do not follow this hierarchical structure are called invalid and may result from BGP misconfigurations or from BGP policies that are more complex and do not distinctly fall into the above classification [6]. Most work in this field (Sec. II) follows the valley free routing principle.

Current relationships inference algorithms attempt to solve the ToR problem either by using heuristic assumptions or by optimizing some aspects of the ToR assignments. Optimization is usually achieved by minimizing the number of paths that violate the valley free routing property [19] while not allowing cycles to be created [5], [16] in the resulting directed relationships graph. Using heuristic assumptions throughout the relationships inference process causes the erroneous ToRs to be spread over all interconnecting ASes links. The optimization models fail to capture the true Internet hierarchy [7] and have a relatively low p2p inference accuracy [22]. The result is that both solutions fail to provide an insight or a bound on the inference errors.

This paper aims to improve on existing methods by providing a near-deterministic inference scheme for solving the ToR problem (we refer to this algorithm as *ND-ToR*). The input for ND-ToR is the Internet *Core*, a sub-graph that consists of the globally top-level providers of the Internet graph and their interconnecting edges with their already inferred relationship types. Theoretically, given an accurate core with no relationships errors, the algorithm *deterministically* infers most of the remaining AS relationships using the AS-level paths relative to this core, without incurring additional inference errors. In real-world scenarios, where the core and AS-level paths can contain errors (due to misconfigurations or measurements mistakes), the algorithm introduces minimal inference mistakes. The core can be approximated in several ways, as described in section III-A, or extracted from public databases. We show that ND-ToR has relaxed requirements from the core, and proves to be robust under changes in its definition, size and density. Since the top-level ASes are a small and stable group, accurately revealing the core members and their mutual types of relationships is fairly easy.

For the remaining set of relationships that cannot be inferred deterministically, a heuristic inference method is deployed. Since this group is relatively small, it is possible to provide a strict bound on the inference error.

The remaining of this paper is organized as follows. Section II discusses related work. Section III provides a detailed description of ND-ToR and discusses methods used to infer the remaining unclassified links. Section IV provides an evaluation of the algorithm. In section V we validate the results of the algorithm by comparing them to the results of some previous major inference algorithms, and section VI concludes the paper.

II. Related Work

Gao's pioneering work [9] was the first to study the AS relationships inference problem. Gao proposed an inference heuristic that identified top providers and peering links based on AS size, which is proportional to its degree (the number of immediate neighbors of a vertex), and the valley-free nature of routing paths. Gao used this heuristic to infer relationship between ASes in the Internet by traversing advertised BGP routes, locally identifying the top provider for each path, and classifying edges (i.e., inferring the relationships represented by the edges) as going uphill to the top provider and downhill afterwards. Xia and Gao [22] later proposed to use partially available information regarding AS relationships in order to infer the unknown relations. A similar idea was more recently suggested by Mühlbauer *et al.* [17]. It is not clear however, that this partial information can be obtained and validated periodically, unlike our suggestion to automatically discover and use the almost constant relationships in the Internet core. Additionally, our sole reliance on the core produces simpler inference rules that are less prone to inference errors.

Following Gao's work, Subramanian *et al.* [19] formally defined the Type-of-Relation (*ToR*) maximization problem that attempts to maximize the number of valid (valley-free) routing paths for a given AS graph. Their approach takes as input the BGP tables collected at different vantage points and computes a rank for every AS. This rank is a measure of how close to the graph core an AS lies (equivalent to vertex coreness [1]), and is heuristically used to infer AS relationships by comparing ranks of adjacent ASes. If the ranks are similar, the algorithm classifies the link as p2p, otherwise as either c2p or p2c.

Battista *et al.* [2] showed that the decision version of the ToR problem (*ToR-D*) is an NP-complete problem in the general case. Motivated by the hardness of the general problem, they proposed approximation algorithms and reduced the ToR-D problem to a 2SAT formula by mapping any two adjacent edges in all input AS-level routing paths into a clause with two literals, while adding heuristics based inference.

Cohen and Raz [5] follow previous works [11], [10] and describe an algorithm that attempts to minimize the number of invalid paths, but in addition, captures the true hierarchal structure of the real Internet. Following the fact that the real Internet graph cannot contain cycles, they defined the Acyclic Type of Relationship (*AToR*) problem as an attempt to maximize the number of valid paths from a given set of routing paths, while keeping the directed graph acyclic.

Dimitropoulos *et al.* [7] addressed a problem in current ToR algorithms. They showed that although ToR algorithms produce a directed Internet graph with a very small number of invalid paths, the resulting AS relationships are far from reality. This led them to the conclusion that simply trying to maximize the number of valid paths (namely improving the result of the ToR algorithms) does not produce realistic results. Later in [6] they showed that ToR has no means to deterministically select the most realistic solution when facing multiple possible solutions. In order to solve this problem, the authors suggested a new objective function by adding a notion of "AS importance", which is the AS degree "gradient" in the original undirected Internet graph. The modified ToR algorithm directs the edges from low importance AS to a higher one. The authors showed that although they have high success rate in p2c inference (96.5%) and in s2s inference (90.3%), the p2p inference success rate (82.8%) is relatively low. Moreover, the authors surveyed some ASes operators and mention that for some of them, the BGP tables, which are the source for AS-level routing paths for most works in this research field, miss up to 86.2% of the true relationships between adjacent ASes, most of which are of p2p type.

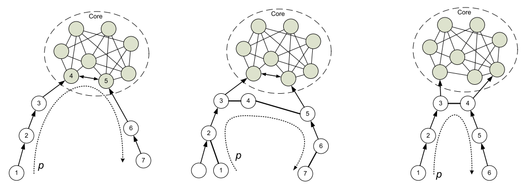
These observations match the evaluation work done in [22], and highly motivate our work, driving us not only to seek an algorithm that better captures the true AS relationships in the Internet while reducing the usage of heuristics for inference, but also to add a different, complementary data source for routing paths, that has the ability to capture much of the missing links.

III. Near Deterministic Inference

The deterministic algorithm receives as input the undirected AS graph and the core graph. Prior to starting the relationships inference algorithm, we infer s2s relationships, since ignoring these relationships might cause proliferation of erroneous inference [6]. We use s2s data collected from [3]. Once classified the two adjacent ASes are united to form a single vertex that inherits the connectivity of both.

Phase 1. All paths that *pass through the core* are classified using the valley-free rule (see Fig. 1(a)), starting with the uphill segment of the path, classifying each edge as c2p, until reaching the core. Inside the core edges are classified as p2p. Downhill links from the core are classified as p2c. Invalid paths are detected when an edge is directed towards the core during the downhill segment.

Since the remaining paths do not traverse the core, they do not provide us with a direct method for classification. However, amongst these, there are paths that partly overlap other paths that traverse the core. Meaning that some of the remaining paths already contain edges that were classified as either c2p or p2c in the first phase of the



(a) Phase 1 (b) Phase 2 (c) Phase 3

Fig. 1. Deterministic ToR inference algorithm

algorithm. We use these edges for the second phase of the algorithm:

Phase 2. For a given path (see Fig. 1(b)), edges that precede a c2p edge must reside in an uphill segment, and be of type c2p. Edges that follow a p2c edge must be in a downhill segment, and be of type p2c. Since this phase uses classified edges in order to classify unclassified edges, it is repeated for all paths that contain unclassified edges, until there are no more edges that can be classified using this method.

To avoid incorrect inferences resulting from transient routing effects or changes in the commercial relationships between ASes, we use voting technique [9] instead of direct relationship inference. Each method "vote" for the ToR it infers, and once the phases are finished, the edge is classified with the type that received a relative votes count that passes a given threshold.

Phase 3. The deterministic phases above fail to classify edges that appear in paths that do not traverse the core, and reside between a c2p edge and a p2c edge (see Fig. 1(c)). Assuming that most c2p and p2c edges are already classified by the deterministic phases, we classify a *single* remaining unclassified edge between a c2p and a p2c edges as a p2p edge. Finally, voting ties between two or more types of relationships are heuristically broken by comparing the k-shell index [4] of the adjacent ASes.

A. Core Graph Construction

Motivated by the need to capture the true global hierarchal structure of the Internet we looked for an accurate global decomposition of the Internet AS-level graph. There have been several attempts to characterize the core of the Internet AS graph [19], [20], [4], [13], [12]. We use three core construction methods, that result in cores that vary in size and density. We analyze the effect that each core has on the classification algorithm.

Tauro *et al.* [20] proposed the *Jellyfish* conceptual model in which they identified a topological center and classified vertices into layers with respect to the center. The authors defined core as a clique of high-degree vertices, and constructed it by sorting the vertices in non-increasing degree order. The first vertex in the core is the one with the highest degree. Then, they examine each vertex in that order; a vertex is added to the core only if it forms a clique with the vertices already in the core. The resulting core is a clique but not necessarily the maximal clique of the graph. We refer to this core as Greedy Max Clique (*GMC*).

Carmi *et al.* [4] indicated that using the popular vertex's degree (which was encouraged by the finding of the Internet's power-law distribution [8]) as an indicator of the vertex's importance can be misleading. The authors presented the new *Medusa* model, that uses a k-pruning algorithm to decompose the Internet AS graph and extract a nucleus (the K_{max}-Core) which is a very well connected globally distributed subgraph. Note that this algorithm extracts a core by looking at the entire graph, unlike GMC that takes a local approach. The properties listed for this model are useful for AS relationship inference, mainly due to the finding that the nucleus plays a critical role in BGP routing, since its vertices lie in a large fraction of the paths that connect different ASes. We refer to this core as k-Core.

The last core we use is constructed from the ASes and interconnecting edges that exhibit p2p relationship using the inference method in [6]. We use the Automated AS ranking provided by CAIDA [3] and constructed a graph that contains all the edges classified as p2p and their adjacent AS vertices. We then selected the largest connected component that contains some of the largest tier-1 ASes, namely AS701 ($UUNET$) and AS7018 ($AT\&T$). We refer to this core as CAIDA Peers (*CP*).

The three core types vary in size and density as an attempt to capture different inference behaviors. Using a small, dense core reduces the probability that a non-top-level AS is wrongfully considered as a top-level AS for all paths that pass through it, thus causing incorrect inferences. However, a small core might miss top-level ASes, thus cause non-valley-free paths. On the other hand, when using a large core, a trace might have several hops in the core. In this case we follow [19] and assume that two ASes may have an "indirect peering" relation, meaning they have p2p relationship through an intermediate AS, such as an exchange point. Traces with more than three hops in the core are considered invalid.

IV. Experimental Results

In this section we evaluate the deterministic algorithm and heuristic phases using data from the first five weeks of 2007. We merged data from RouteViews (RV) [21] and DIMES [18] projects to maximize the size of the resulting AS topology. The data set consists of over 24,000 AS vertices and approximately 58,000 undirected links. Approximately 42% of the edges exist only in RV, about 12% exist only in DIMES paths and the remaining 46% of the edges exist in both RV and DIMES.

A. Voting Threshold

Validation of the voting technique and determining a threshold value is achieved by analyzing the distribution of votes to inference types. For each edge we calculated the number of p2c votes out of the total number of votes (i.e., the p2c votes ratio). Fig. 2 shows the number of edges for each p2c ratio. Clearly, the vast majority of the edges are uniquely classified as either p2c or c2p. This holds when running ND-ToR on longer time frames.

Looking at the data backing up this graph, we see that on average over 94% of the edges have votes for exactly

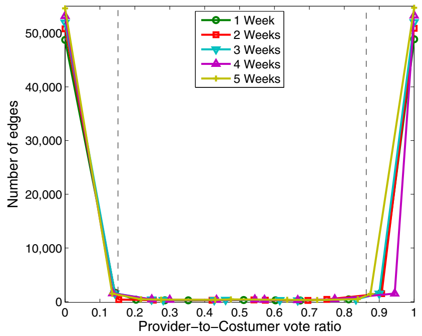


Fig. 2. Types-of-Relations voting distribution

one relationship type, and almost 99% of the edges have over 80% of the votes casted for a single relationship type, which provides a very high level of confidence for this selected type. Thus, a threshold value of 0.85 covers almost 99% of the edges, and leaves approximately 1% of the edges to be classified using heuristic methods, or remain unclassified.

B. Sensitivity Analysis

Since the construction of the core graph is an important building block for ND-ToR, we evaluate the effect that the core has on the inference process. First, the stability of the resulting core and the ability to algorithmically construct it are analyzed. Then, the overall algorithm performance over consecutive weeks is examined, followed by an evaluation of the optimal core size, i.e., a core that results in a minimal inference mistake while achieving a high classification percentage. Finally, the sensitivity of ND-ToR to errors in the core is validated, by randomly replacing core vertices.

We first validate that it is possible to consistently discover a stable set of core ASes. Fig. 3 shows the ASes in k-Core and GMC (AS numbers are mapped to consecutive indexes, k-Core is marked as '$\cdot$' and GMC is marked as '$\circ$'), using data from the first five weeks of 2007.

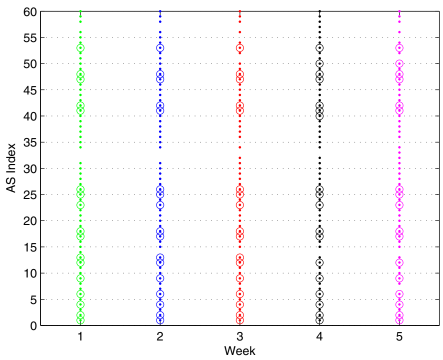


Fig. 3. Stability of core vertices, k-Core marked with '$\cdot$', GMC marked with '$\circ$'

Several aspects of the different cores are revealed by looking at the ASes comprising the cores. The first observation is that both algorithms produce a relatively stable core, with minor variations in the actual ASes that are selected in each core. The second observation (A table with the data is omitted due to space limitation) is that while most GMC ASes have relatively high degree and are concentrated in North America (except 2 in Europe), the ASes in k-Core have more diverse geographic and degree distributions. Namely, the k-Core decomposition has a more global view of the Internet AS-level graph. The last observation is that GMC core is included in k-Core; the global decomposition performed by the k-Core algorithm manages to capture the dense core discovered using GMC.

Next, we examined the result of running ND-ToR on the topology of each of the first five weeks of 2007, each time with a different core type. As expected, most of the AS relationships are inferred in phase 1 of the deterministic algorithm using paths that traverse the core. These paths comprise a large percentage of all available paths, ranging from over 95% for k-Core and CP to 88% for the smaller GMC core.

Table I shows the structure of the different core types used and the effect it has on the deterministic inference algorithm. It shows that the smallest GMC core results in the lowest deterministic inference percentage while the largest CP core have the highest percentage. This is the result of the larger cores having more paths that traverse them, therefore can be deterministically inferred. k-Core provides an excellent overall inference percentage (over 95% deterministically inferred). Additionally, the results are stable over the measured five weeks period.

TABLE I
THE EFFECT OF INPUT CORES' VERTICES AND EDGES ON DETERMINISTIC INFERENCES

Core	Week→	1	2	3	4	5
k-Core	Vertices	57	56	54	58	54
	Edges	2260	2198	2076	2344	2134
	Classified	95.59%	95.76%	95.34%	95.24%	94.22%
Greedy Max Clique	Vertices	17	17	17	18	17
	Edges	272	272	272	306	272
	Classified	89.64%	89.87%	89.77%	89.62%	88.87%
CP	Vertices	1067	1053	1068	1056	1087
	Edges	6158	6110	6012	5844	6138
	Classified	98.29%	98.55%	98.45%	98.0%	97.39%

Although CP core seems to result in the best overall performance, constructing the CP core revealed that only a few p2p edges out of the approximately 6,000 edges were not a part of the largest connected component. This suggests that CAIDA incorrectly infers AS relationships as p2p, since it is highly unlikely that almost all p2p edges are connected. This causes a bias, resulting in more inference errors.

Table II shows that less than 6% of the edges were differently classified using two cores in each week, and the difference between k-Core and GMC is much smaller. This shows that ND-ToR results are relatively consistent regardless of the input core.

In order to find the best core size, we run ND-ToR with a growing core size starting at four vertices. We do this for two of our core types – k-Core and GMC, using the first

TABLE II
PERCENTAGE OF EDGES THAT CHANGE CLASSIFICATION COMPARING DIFFERENT CORE TYPES

Cores | Week→	1	2	3	4	5
k-Core - GMC	1.77%	1.66%	1.58%	1.8%	1.64%
k-Core - CP	5.94%	5.89%	5.84%	5.7%	5.81%
GMC - CP	3.53%	3.45%	3.37%	3.34%	3.51%

week of 2007. We start with the highest degree vertices and add vertices in a non-increasing degree order. Using k-Core, we first add vertices from the $K_{max} - Core$ and then proceed to shells with lower indexes.

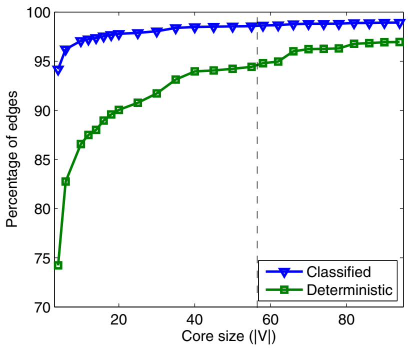


Fig. 4. Percentage of classified and deterministically classified links using different input core size

Fig. 4 shows the robustness of ND-ToR relative to the size of the input core (GMC graph omitted due to lack of space). The vertical dashed line marks the true core size. It shows that for more than 20 vertices in the core, ND-ToR classification success do not significantly change, while the number of deterministically classified edges increases. However, this increase comes with an increase in the percentage of non-valley-free paths as shown in Fig. 5. This implies that the core must be kept small enough to decrease the number of invalid paths.

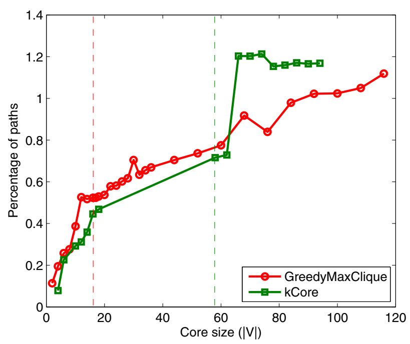


Fig. 5. Percentage of non-valley-free paths when increasing input core size

Overall we showed that it is possible to create a stable core as input to the proposed algorithm. ND-ToR results are consistent over time and various cores. Additionally, a core containing approximately 20 top-level ASes is sufficient to obtain excellent inference results.

C. Time Aggregation Analysis

We wish to find a time frame for which ND-ToR captures best the relationships between ASes. A short time frame results in a fast running algorithm but might miss AS links and AS paths, especially in the DIMES data. This results in a low vote count, possibly decreasing the success of ND-ToR. On the other hand, a long time frame captures two effects that can also cause a decrease in inference success: 1) commercial relationships can change and complex routing behaviors may occur over long durations, and 2) possible measurements mistakes can pile up and skew the results.

We applied ND-ToR on an increasing time frame. We started with the first day of 2007 and aggregated single days until the end of the first week (for this daily analysis we used three RouteViews files a day). Then, we aggregated a week at a time, until reaching 10 consecutive weeks. DIMES provides approximately 1.5M non-unique tracroutes each day, reaching over 100M traceroutes for the 10 weeks period. RouteViews provides approximately 1.2M *unique* paths regardless of the time frame used.

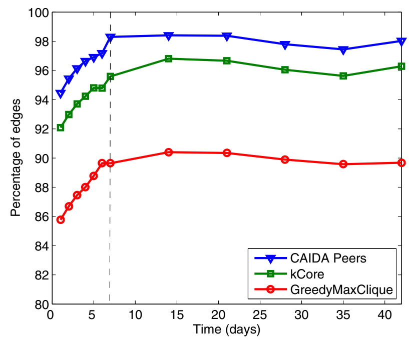


Fig. 6. Percentage of deterministically classified links over an increasing time frame

The percentage of deterministically inferred edges over the aggregated time frame is shown in Fig. 6. Using data from a single week (marked as the vertical dashed line) results in over 90% of the edges being classified for all core types, having CP obtaining the best percentage and GMC the worst. This is directly related to the size of the core, since a larger core results in more paths that traverse through it, yielding more deterministically inferred relationships.

Finally, we looked at the consistency of the inference results over the time frame by comparing edges that are classified in both time frames. We found that over 98% of the inferences remain constant between consecutive time frames. This suggests that there are only a few commercial relationships that change over time. Short-term routing changes have very little effect, since they statistically "disappear" as the more common routes become dominant over time.

D. Non-Deterministic Inferred Relationships

Edges that the deterministic algorithm fails to classify are classified using the two heuristic-based inference methods – breaking voting ties and inferring p2p relationships. To break voting ties the algorithm compares adjacent AS degrees (similar to [9]) and infers the relationship between them as p2p if the degrees ratio is between 0.8 and 1.2, or p2c otherwise (marking the provider as the AS with the higher degree). When using k-Core the algorithm compares the k-Shell index of two adjacent ASes, and infers the relationship to be p2p if the two ASes have the same k-Shell index, or p2c otherwise (marking the provider as the AS with the higher k-Shell index). This heuristic is an attempt to overcome the problem in degrees comparison, occurring since p2p links often exist between ASes that have very different degrees [14]. We, however, noted very little difference between the two heuristics.

E. Robustness to Core Errors

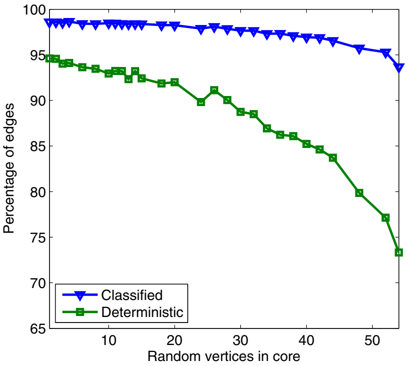


Fig. 7. Percentage of classified and deterministically classified links when increasing the number of random ASes in k-Core

Estimating the accuracy and robustness of ND-ToR is done by intentionally increasing the mistake in the core. This is performed by randomly replacing ASes in the core and examining the change in the number of inferred relationships. We start by replacing one core AS with one random AS (that is connected to at least one of the remaining core ASes) and gradually replace more ASes until we have a core that consists of completely random but still connected ASes. Fig. 7 shows the percentage of classified edges and deterministically classified edges using k-Core.

Interestingly, while ND-ToR's performance decreases as we increase the randomness of the core, the overall degradation is not as high as one would expect. However, Fig. 8 shows a rising trend of the percentage of unclassified, p2p and tie-breaking heuristic edges as we inject errors to the core. As more errors are injected, ND-ToR needs to use more heuristics. Particulary, when there are approximately 50% random vertices in the core, the effect of the increasing mistake becomes more noticeable. However, even with a completely random core (note that randomness is limited by the requirement that the core remains connected), the overall heuristically inferred edges account for less than 20% of all edges.

Similar behavior to the one depicted in Fig. 7 and Fig. 8 was obtained for the GMC core. These results indicate that although ND-ToR seems quite robust to the mistake

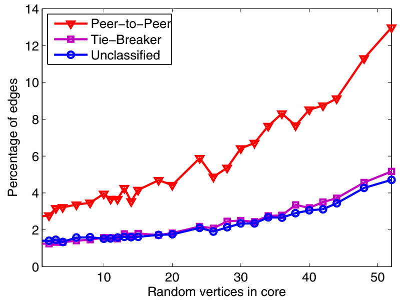


Fig. 8. Percentage of p2p, heuristically classified and unclassified links when increasing random ASes in the core

in the core, it is still significantly affected once there are more than 50% incorrect core vertices.

F. Data Source Coverage

Analysis of the ToR of edges discovered by DIMES, is done by looking at the ToR distribution of edges that appear only in DIMES, and are not seen in the BGP routing tables. Table III shows the distribution of ToR that were inferred using ND-ToR on data from the first week of 2007. While on average p2p relationships comprise 4-5% of the total number of edges, they go up to around 9% (using k-Core) and almost 12% (using GMC) of the edges that appear only in DIMES. Moreover, approximately 40% of the p2p edges inferred by ND-ToR, do not appear in the RV tables. This strengthen the claim that DIMES significantly improved the ability to detect p2p links between ASes, mainly since DIMES agents are diversely spread over the Internet and contribute AS links that are not seen by the RouteViews either since they are not propagated due to BGP rules to RV collectors or even not published by the BGP protocol at all.

TABLE III
ND-ToR's inference types distribution

Core	Edges	P2C	P2P	S2S	Unres
k-Core	all	107936 (94%)	4762 (4.1%)	474 (0.4%)	1570 (1.5%)
	DIMES only	12348 (88.6%)	1240 (9%)	0 (0%)	340 (2.4%)
GMC	all	104720 (90.5%)	7655 (6.6%)	474 (0.4%)	2901 (2.5%)
	DIMES only	11886 (85.3%)	1598 (11.5%)	0 (0%)	444 (3.2%)

V. Validation

Validating the accuracy of the results was done by comparing ND-ToR to other perviously suggested algorithms. While implementing these algorithms we revealed several problems that prevented us from using some, or limited our ability to use others, as described below.

The PTE [22] algorithm requires partial relationships data that should be manually collected. Finding accurate relationships data is an extremely difficult task, becoming almost impossible when the relationships for a specific time frame in the past is needed. Hence, this algorithm was not selected for comparison.

A different problem arises when executing the heuristic phase of the AToR [5] algorithm, used to infer p2p and s2s relationships. This phase selects a set of candidate edges, and for each assignment of p2p and s2s relationships to a permutation of the candidate edges, it examines all the paths to check which assignment causes the smallest number of non-valley-free paths. This results in an almost impossibly long run-time given a large set of input paths. Therefore, only the p2c assignments of the AToR algorithm are compared. Another algorithm used for comparision, is the one suggested by Battista *et al.* [2], that was the first to reduce the ToR problem to MAX2SAT. These two algorithms are referred to as AToR and BPP respectively, and are used to validate only p2c relationships.

CAIDA publishes the results of the inference method described in [6] in the automated AS ranking web site [3]. While this data is easily accessible, we can only use it to infer links visible from RoutViews, since they do not use DIMES data. This algorithm is referred to as CAIDA.

Finally, the results are also validated against the first AS relationship inference algorithm, which was proposed by Gao [9]. This algorithm is referred to as GAO.

We start with validating the deterministic phases of ND-ToR. This is achieved by comparing only the deterministically inferred edges (using the three different core types) with the results of CAIDA, over the first six weeks of 2007 (although incomplete, CAIDA's inference results are the most easily accessible for a long duration in the past, and are sufficient for this validation). Fig. 9 shows the edges that are identically inferred by the deterministic algorithm and CAIDA (percentage is taken from the edges that are inferred by both algorithms). It shows that for all cores and any time frame, the deterministic edges inferred by ND-ToR and all of the edges inferred by the heuristic methods of CAIDA agree for over 92% of the edges.

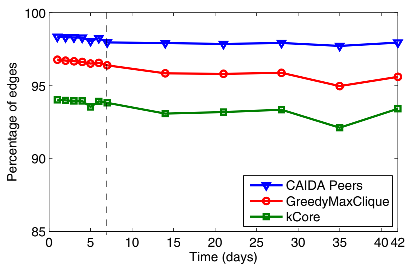


Fig. 9. Percentage of deterministically inferred links matching CAIDA (calculated out of links that are inferred by both algorithms)

Table IV provides a comparison of the inference results obtained from running the different algorithms using data comprised of AS-level paths from both RV and DIMES for the first week of 2007. CAIDA inference is an exception since the inferred relationships were downloaded directly from the CAIDA's web site [3], thus does not include DIMES links. The empty slots in the table are due to fact that BPP and AToR are used only for p2c inferences, as discussed above. Note that p2c and c2p inferences are united into a single column named $P2C$.

The table shows the number of edges classified for each ToR class in each of the different algorithms. All algorithms, except CAIDA, are provided with the same AS paths list. While ND-ToR and GAO filter edges from the resulting topology, BPP and AToR do not use any filtering, and infer almost all of the edges given in the ASes paths list, resulting in approximately 50% more edges. On the other hand, CAIDA uses only BGP data, thus it has less edges in its topology.

TABLE IV
COMPARISON OF INFERENCE RESULTS

Algorithm	Criteria	P2P	P2C	S2S
GAO	all edges	6576	102258	6020
CAIDA	all edges	6158	88381	474
BPP	all edges		150012	
AToR	all edges		150550	
ND-ToR *k*-Core	all edges	4762	107936	474
	$= GAO$	1830	98492	48
	$= CAIDA$	2056	85706	474
	$= BPP$		102971	
	$= ATOR$		105737	
	$= all$	1172	82102	48
	$\neq$ all	720	59	0
ND-ToR Greedy Max Clique	all edges	7655	104720	474
	$= GAO$	3488	96668	48
	$= CAIDA$	3005	84974	474
	$= BPP$		100968	
	$= ATOR$		102082	
	$= all$	1771	81398	48
	$\neq$ all	1255	85	0

TABLE V
COMPARISON OF INFERENCE DISTRIBUTION

		ND-ToR *k*-Core			ND-ToR GMC		
		P2P	P2C	S2S	P2P	P2C	S2S
	Total	4762	107936	474	7655	104720	474
GAO	P2P	1830	3964	0	3488	1915	0
	P2C	991	98492	213	1899	96668	213
	C2P	983	288	213	1908	266	213
	S2S	925	4901	48	315	5513	48
	Unres.	33	291	0	45	358	0
CAIDA	P2P	2056	3884	0	3005	2313	0
	P2C	573	85656	0	699	84974	0
	C2P	566	1225	0	704	1491	0
	S2S	0	0	474	0	0	474
	Unres.	1567	17171	0	3247	15942	0
BPP	P2C	2362	102972	237	3780	100968	237
	C2P	2367	4547	237	3840	3311	237
	Unres.	33	418	0	35	441	0
AToR	P2C	2376	105737	237	3811	102082	237
	C2P	2368	2130	237	3827	2580	237
	Unres.	18	69	0	15	58	0

We compare the distribution of inference types of GAO and CAIDA algorithms to ND-ToR, since all three provide us information on all types. GAO infers significantly more S2S edges than CAIDA and ND-ToR (recall that ND-ToR uses the s2s inference of CAIDA). It seems unreasonable that the high s2s inference percentage of GAO is correct, since an accurate s2s inference is almost impossible without using IRR databases, which is not fed to GAO's algorithm in our validation process. Indeed, GAO only identify correctly 10% of CAIDA's s2s links.

The inference of p2c edges is rather the same among the algorithms. Although it seems that CAIDA results in the lowest match to ND-ToR, the reason for this is the relatively low number of edges in CAIDA's topology. ND-ToR agrees on over 95% of the p2c edges with each of the examined algorithms, regardless of the input core used (percentage is taken from the minimal number of edges between the compared algorithms). All five algorithms agree on over 95% of the p2c inferred edges. In addition, ND-ToR disagrees with the compared algorithms on only 0.05% of the p2c edges. This proves the correctness of the mostly deterministic p2c inferences done by ND-ToR, and the robustness of the results to the input core used.

Unlike p2c edges that exhibit unity among the algorithms, p2p edges raise some conflicts between the algorithms. The number of p2p edges inferred by GAO, CAIDA and ND-ToR seems to be relatively similar, where ND-ToR slightly differ, depends on the input core used. However, when all are compared we can see that the algorithms agree on only approximately 21% of the p2p edges inferred by ND-ToR. However, ND-ToR seems to disagree on only 15% of the p2p edges that CAIDA and GAO agree on. It is interesting to see that using k-Core, p2p inferences made by ND-ToR are more similar to CAIDA (43%) than to GAO (38%), while using GMC, ND-ToR results in p2p inferences that are closer to GAO (45%) than to CAIDA (39%). This can be attributed to the fact that GAO uses very local-based heuristics, similar to the core resulting from the local properties of GMC, while CAIDA takes a more global approach, which is closer in spirit to that of k-Core.

Table V shows the correlation of inference types comparing ND-ToR and the reference algorithms. The table shows, for each of the types inferred by ND-ToR (using k-Core and GMC), the number of edges inferred by the reference algorithms by their corresponding types. It shows that although the majority of edges are inferred the same using ND-ToR (with both cores) and the reference algorithms, there is no distinct agreement among the algorithms (except s2s with CAIDA, since these were extracted from CAIDA's data). Even for p2c, that have a very distinct agreement among all algorithms, there is a small number of edges (up to 0.3% with GAO, 1.5% with CAIDA, 4.3% with BPP and 2.6% with AToR) that are inferred in the wrong "direction" (i.e., the provider in the first algorithm is a customer in the second and the customer in the first is a provider in the second).

Additionally, the table shows that the overall percentage of unresolved edges is very small (less than 0.7%) for all algorithms except CAIDA (that has less edges, since it uses only BGP paths). Among CAIDA's inferences, the p2p relationship has the highest percentage of unresolved edges (compared to ND-ToR, over 32% using k-Core and 42% using GMC). This strengthen our claim that most of the links missing from BGP paths are of type p2p.

VI. Conclusion

The common weakness of previously proposed AS relationships inference algorithms is their lack of guarantee on inference errors introduced during the process. This work improves on existing methods by providing a near-deterministic algorithm that, given a classified error-free input core, does not introduce additional inference errors. We investigate various input cores and show that the proposed algorithm provides accurate inferences that are robust under changes in the core's size and creation technique. We show that a core containing as little as 20 almost fully-connected ASes is sufficient for good inference results. Additionally, we show that heuristic methods can still play an important role in inferring the remaining relationships. Using data collected from a single week (containing approximately 1.2M BGP paths and over 10M DIMES AS-level traceroutes), ND-ToR runs for only about 2 hours and yields over 95% deterministically inferred relationships.

References

[1] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. k-core decomposition: a tool for the analysis of large scale internet graphs, 2005.

[2] G. D. Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems. Technical Report RT-DIA-73-2002, Dipartimento di Informatica e Automazione, Universita di Roma Tre, 2002.

[3] CAIDA. Automated Autonomous System (AS) ranking. Research Project. http://as-rank.caida.org.

[4] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir. A model of Internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences USA (PNAS)*, 104(27), July 2007.

[5] R. Cohen and D. Raz. Acyclic type of relationships between autonomous systems. In *IEEE INFOCOM*, 2007.

[6] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley. AS relationships: Inference and validation. *ACM SIGCOMM Computer Communications Review*, 37:2007, 2006.

[7] X. Dimitropoulos, D. Krioukov, B. Huffaker, kc claffy, and G. Riley. Inferring as relationships: Dead end or lively beginning? *LNCS*, 3503:113, 2005.

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, 1999.

[9] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. on Networking*, 9(6):733–745, 2001.

[10] L. Gao, T. Griffin, and J. Rexford. Inherently safe backup routing with BGP. In *INFOCOM*, pages 547–556, 2001.

[11] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Measurement and Modeling of Computer Systems*, pages 307–317, 2000.

[12] Z. Ge, D. Figueiredo, S. Jaiwal, and L. Gao. On the hierarchical structure of the logical internet graph. In *SPIE ITCOM*, Aug. 2001.

[13] R. Govindan and A. Reddy. An analysis of internet inter-domain topology and route stability. In *INFOCOM*, pages 850–857, 1997.

[14] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and impact. In *NSDI*, 2007.

[15] G. Huston. Interconnection, peering, and settlements. In *INET*, San Jose, CA, USA, June 1999.

[16] S. Kosub, M. G. Maaß, and H. Täubig. Acyclic type-of-relationship problems on the internet. In *The 3rd Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN'06)*, volume 4235 of *LNCS*, pages 98–111. Springer-Verlag, July 2006.

[17] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-topology model that captures route diversity. In *ACM SIGCOMM*, pages 195–206, Aug. 2006.

[18] Y. Shavitt and E. Shir. DIMES: let the internet measure itself. *ACM SIGCOMM Computer Communications Review*, 35(5):71–74, 2005.

[19] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *INFOCOM*, New York, NY, USA, Apr. 2002.

[20] L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the Internet topology. In *Global Internet*, Nov. 2001.

[21] University of Oregon Advanced Network Technology Center. Route Views Project, http://www.routeviews.org/.

[22] J. Xia and L. Gao. On the evaluation of AS relationship inferences. In *IEEE Globecom*, Dallas, TX, USA, Nov. 2004.