NPS-CS-15-001

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

GRAPH REDUCTION FOR EMULATED NETWORK EXPERIMENTATION

by

Erik Rye
Justin P. Rohrer

March, 2015

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
04-31-2015	Technical Report	01-01-2015 to 04-31-2015

4. TITLE AND SUBTITLE

GRAPH REDUCTION FOR EMULATED NETWORK EXPERIMENTATION

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Erik Rye, Justin P. Rohrer

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-CS-15-001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.IRB Protocol Number:N/A.

14. ABSTRACT

Network researchers and operators often turn to emulation and simulation for testing and experimentation.Obtaining topologies that reflect the graph characteristics of the Internet, while of small enough order to emulate or simulate on commodity hardware, however, is a difficult undertaking.In this work, we reexamine a previous study devoted to generating Internet-like topologies by reducing Autonomous System-level Internet instances to a more manageable scale.In addition to replicating the original experiment using Routeviews data from 2001, we extend the prior work's methodology to more current data and to another data set compiled by the Center for Applied Internet Data Analysis.Finally, we introduce a new Internet graph reduction method, and examine its performance on both data sets.

15. SUBJECT TERMS

Graph reduction, graph algorithms, network emulation, Internet modeling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Justin P. Rohrer
Unclassified	Unclassified	Unclassified	UU	29	19b. TELEPHONE NUMBER *(include area code)* 831-656-3196

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Ronald A. Route
President

Douglas A. Hensler
Provost

The report entitled "Graph Reduction for Emulated Network Experimentation" was prepared for the Naval Postgraduate School based on the results of an independent study conducted by the authors.

Further distribution of all or part of this report is authorized.

This report was prepared by:

Erik Rye
Capt, USMC

Justin P. Rohrer
Research Assistant Professor

Reviewed by:

Peter J. Denning, Chairman
Department of Computer Science

Released by:

Jeffrey D. Paduan
Dean of Research

Table of Contents

List of Acronyms and Abbreviations

BGP Border Gateway Protocol

RIB Router Information Base

CAIDA Center for Applied Internet Data Analysis

AS Autonomous System

CRVE contraction of a random vertex/edge

CRE contraction of a random edge

DRE deletion of a random edge

DRV deletion of a random vertex

DRVE deletion of a random vertex/edge

EBFS exploration by Breadth-First Search

EDFS exploration by Depth-First Search

DHYB deletion-hybrid

KDD k-core - DRVE - DRE

KKD k-core - k-deletion - DRE

Abstract

Network researchers and operators often turn to emulation and simulation for testing and experimentation. Obtaining topologies that reflect the graph characteristics of the Internet, while of small enough order to emulate or simulate on commodity hardware, however, is a difficult undertaking. In this work, we reexamine a previous study devoted to generating Internet-like topologies by reducing Autonomous System-level Internet instances to a more manageable scale. In addition to replicating the original experiment using Routeviews data from 2001, we extend the prior work's methodology to more current data and to another data set compiled by the Center for Applied Internet Data Analysis. Finally, we introduce a new Internet graph reduction method, and examine its performance on both data sets.

1 Introduction

Advances in the performance of computer hardware over the last decade, coupled with the ability to virtualize and emulate router operating systems, have provided researchers and operators with an invaluable tool. [1] Rather than physically building time- and resource-intensive network topologies, individuals can now design and create complex networks in an emulated testbed to analyze and test the performance of designs and protocols. For researchers interested in designing emulated networks that mimic the behavior observed on the Internet, however, hardware limitations are quickly realized. The question has shifted to how to best create network topologies with the graph theoretical properties that match those observed in Internet graphs as closely as possible.

In order to obtain realistic, Internet-like topologies, two distinct types of methodologies have been described. The first is a constructive approach - beginning with a small graph, then adding nodes and edges until an instance of the desired order has been obtained. Many methods have been proposed to accomplish this, including the Barabási-Albert, Waxman, Inet, and LocGen generation models [2], [3], [4], [5]. A second approach is a reductive one - beginning with an initial Internet instance, and reducing it until a graph of the desired order is obtained, retaining the properties that an Internet instance *of that order* would have. This approach was studied in detail in [6], and we undertake a reexamination of the graph reduction methods introduced by the authors in order to evaluate their viability with current Internet data.

2 Methodology

In this work, we consider two data sets over two time frames. The first data set we evaluate is the Border Gateway Protocol (BGP) Router Information Base (RIB) archive located at

`http://archive.routeviews.org`, which is also the data set that the authors of [6] used in their work. The second is the Autonomous System (AS) relationship data set compiled by the CAIDA, located at `http://data.caida.org/datasets/`. Though these two data sets was gathered by different methods, both can be used to construct an AS-level graph of the Internet.

In order to recreate and validate the work of [6], we examine the period between May 2001 and January 1998, using the graphs obtained from the Routeviews BGP RIB tables. Specifically, we begin with the Internet instance obtained from 7 May 2001 at 1800 as our initial instance, and reduce to a graph equivalent to roughly the order of the 24 January 1998 Internet instance. To determine whether the reduction mechanisms the authors of [6] propose remain valid with more current data, we also examine Routeviews BGP RIB data from 1 December 2014 and reduce to graphs of the order of the Internet instance occurring on 1 January 1998, a span of almost 17 years.

To broaden the results we obtained during our study, we also consider CAIDA data from the same time period. We begin our CAIDA reductions from initial Internet instances obtained on 1 May 2001 and 1 December 2014, reducing these to graphs of the same order as the Internet instance inferred by CAIDA from 1 January 1998.

With our initial Internet instances and reduction endpoints, we also created graphs from intermediate Internet instances in order to compare our graph reduction methods to the actual Internet's evolution during the reduction process for certain metrics. In general, we have selected our intermediate graphs six months apart for the reductions from 2014 to 1998, and four months apart for the reductions from 2001 to 1998.

In order to reproduce and expand upon the experiments conducted by the authors of [6], we now describe them metrics used for comparing our reduced graphs to the Internet instances of the same order. We study three in particular that were studied by previous work, and one that was not.

1. Average degree of the reduced graphs.
2. A spectral analysis of the 100 largest eigenvalues of the normalized adjacency matrices of the furthest reduced graphs.
3. A *hop-plot*, or plot of the CDF of vertex-pairs against the number of hops these pairs are along a geodesic in the reduced graph.
4. The sorted degree-distribution of the reduction endpoint graphs versus the degree distribution of the target Internet instance.

Finally, we introduce the graph reduction methods used in our work. These can broadly be classified into three different categories: contraction, deletion, and exploration.

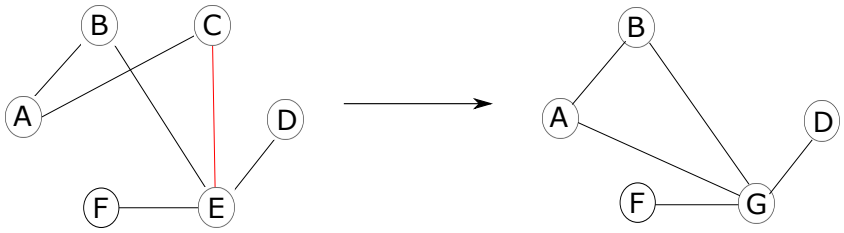

Figure 1: A reduction by the CRE method

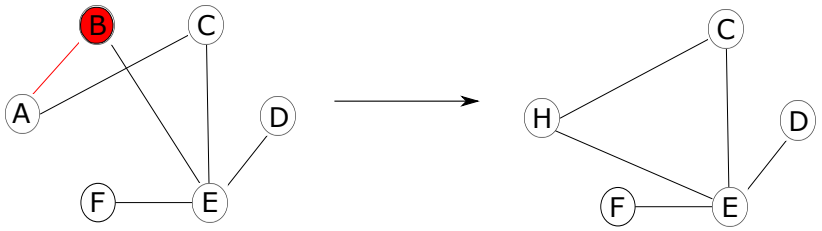

Figure 2: A reduction by the CRVE method

2.1 Contraction

We consider two different types of contraction methods - first, *CRE* [6]. In CRE, an edge in the graph is selected with a uniform probability. The vertices incident to this edge are contracted into one vertex, which now has the union of edges incident to the original two vertices incident to itself (minus the edge that has been contracted). We demonstrate CRE in figure 1.

The *CRVE* [6] method first selects a vertex in the graph with a uniform probability, then chooses an edge incident to that vertex with a uniform probability. The endpoints of this edge are then contracted into a single vertex, which has incident to itself the union of edges incident to the original two vertices minus the edge that has been contracted. A small example of CRVE is given in figure 2.

In both CRE and CRVE, the contraction process is repeated until a graph of the order of the reduction target is obtained.

2.2 Deletion

Within the deletion category, we employ three distinct methods and a hybrid of two of these methods. The first method, *DRV* [6], involves selecting a random vertex with a uniform probability, then removing this vertex (and its incident edges) from the graph. After each vertex removal, a check is done to ensure that resulting graph is still connected. If a disconnected graph has been produced, the largest connected component is retained, while all other components are removed. This process is repeated until we have obtained a re-

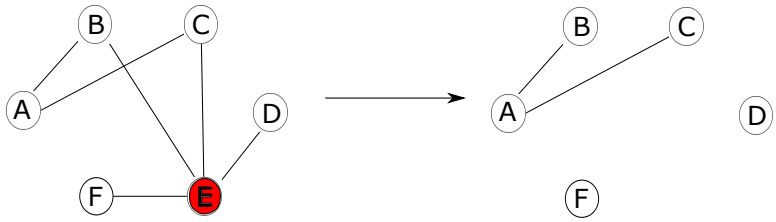

Figure 3: A reduction by DRV. The largest connected component is retained

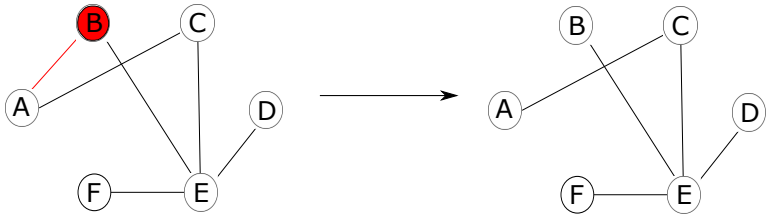

Figure 4: A reduction by DRVE

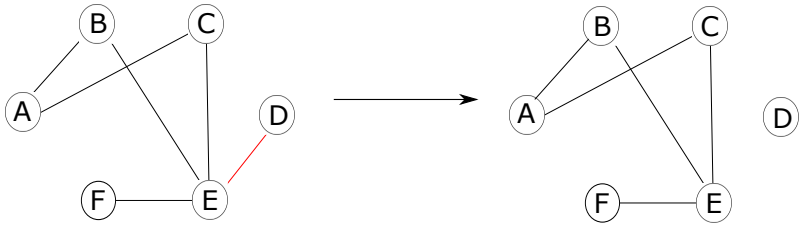

Figure 5: A reduction by DRE

duced graph of approximate order to the Internet instance we are reducing to. Equality of order between the reduced graph and the reduction endpoint may not be realized, as a disconnection and the subsequent removal of disconnected components may reduce the graph beyond the number of nodes we desire; by checking for disconnections after each vertex removal, we obtain a comparable number of vertices in the reduced graph to the final Internet instance. A graphical example of DRV is given in figure 3.

Second, the *DRVE* [6] method begins by first choosing a vertex at random from the original Internet instance with a uniform probability. Then, an edge incident to this vertex is selected with a uniform probability, and removed from the graph. Again, we check for disconnections following each edge removal, and discard all but the largest connected component of the resultant graph. This process is repeated until we have obtained a reduced graph of comparable order to the target Internet instance. Figure 4 demonstrates a reduction by DRVE.

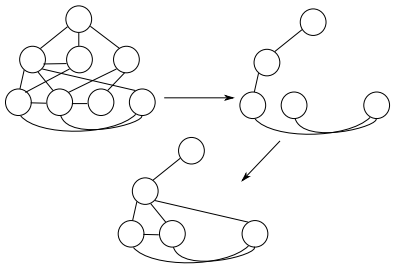

Figure 6: The original graph, a DFS tree rooted at the top node, and the resultant reduced graph

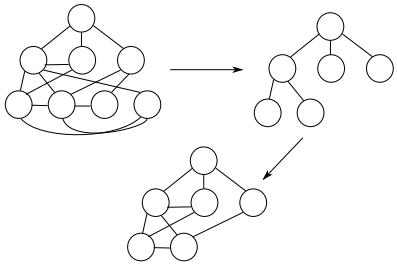

Figure 7: The original graph, a BFS tree rooted at the top node, and the resultant reduced graph

The third method, *DRE* [6], involves selecting a random edge with uniform probability, and removing it from the graph. As in DRV and DRVE, following the removal of each edge, all but the largest connected component are removed if a disconnections have resulted from the edge removal. This process is repeated until we have reached the limit of our reduction. An example of one edge deletion in DRE is given in figure 5.

Finally, we examine a set of *deletion-hybrid (DHYB)* [6] methods, which are hybrids of DRE and DRVE. Specifically, each DHYB-$.p$ method chooses DRVE with a probability of $.p$ and DRE with a probability of $(1-.p)$. Thus, when $p = 1$, this is equivalent to DRVE, and when $0 = p$, is equivalent to DRE. In our work, as in [6], we examine values of p between 0 and 1 in one-tenth intervals. For simplicity, we label these methods without the decimal point in Section 3.

2.3 Exploration

We consider two distinct methods within the exploration category - *exploration by Depth-First Search (EDFS)* and *exploration by Breadth-First Search (EBFS)* [6]. In EDFS, a vertex in the initial Internet instance is selected randomly with a uniform probability. From that vertex, the Depth-First Search algorithm [7] is run until the desired number of vertices have been discovered. The result is the induced subgraph of the original Internet instance induced by the node set of the Depth-First Search tree. A toy example of an 8-vertex graph reduced to a 5-vertex graph by EDFS is given in figure 6.

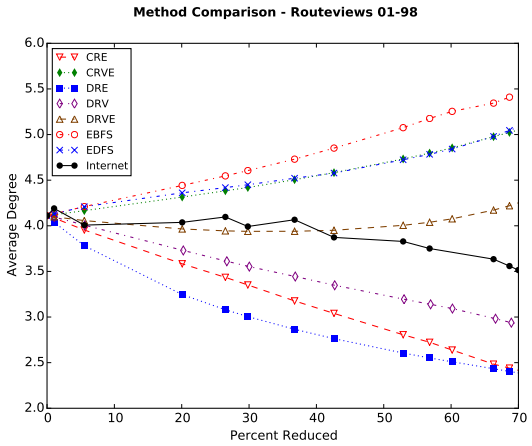

Figure 8: Average degree performance of non-DHYB methods - Routeviews 2001-1998

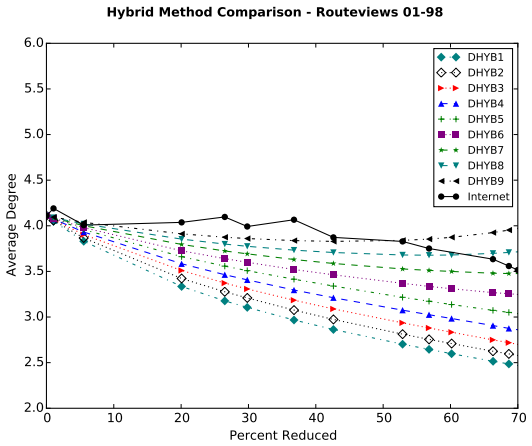

Figure 9: Average degree performance of DHYB methods - Routeviews 2001-1998

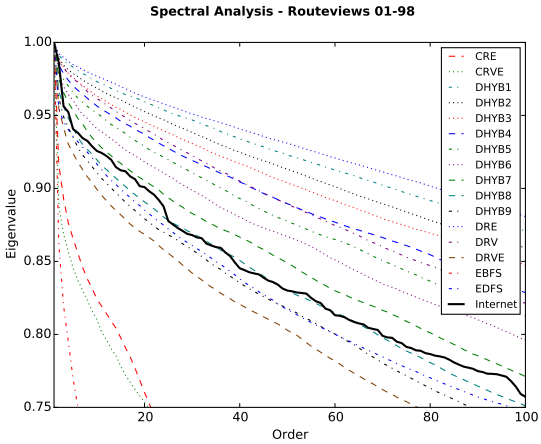

Figure 10: Spectra of reduction methods versus 24 January 1998 Internet graph - Routeviews 2001-1998

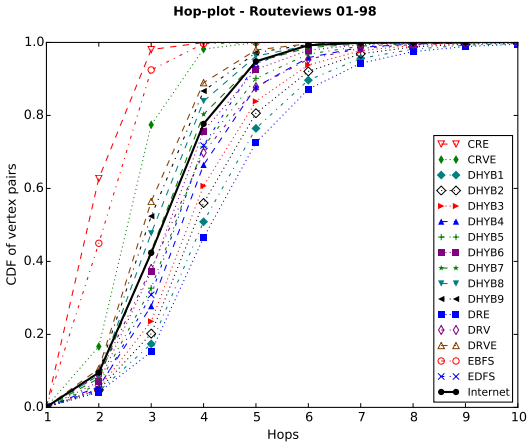

Figure 11: Hop plot of reduction methods versus 24 January 1998 Internet graph - Routeviews 2001-1998

In a similar fashion, EBFS begins by selecting a random start vertex uniformly, and running the Breadth-First Search algorithm [7] until the number of vertices desired has been reached. The node set of the Breadth-First Search tree is then used to induce a subgraph of the original Internet instance, which is the resultant graph obtained. Another small example of an 8-vertex graph reduced to a 6-vertex graph is given in figure 7.

3 Results

We divide our results into four data set and time frame components to analyze individually.

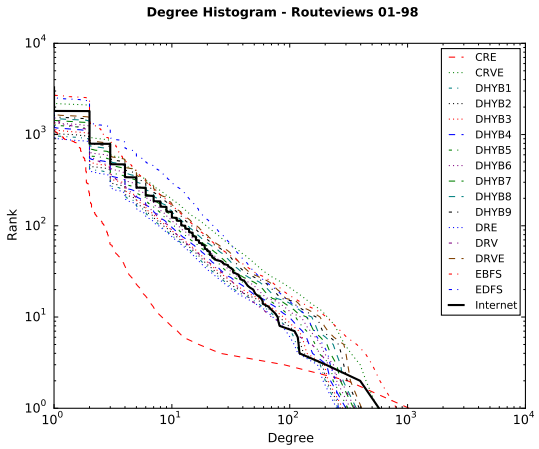

Figure 12: Degree histogram of reduction methods versus 24 January 1998 Internet graph - Routeviews 2001-1998

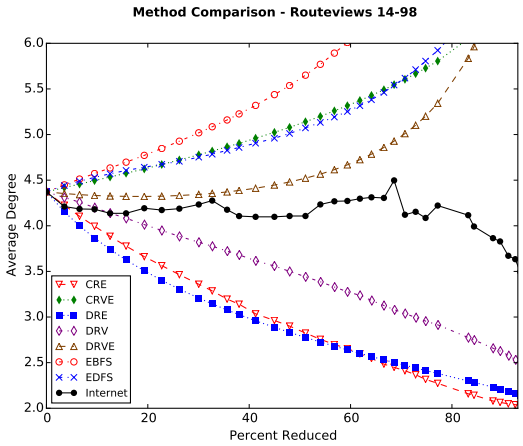

Figure 13: Average degree performance of non-DHYB methods - Routeviews 2014-1998

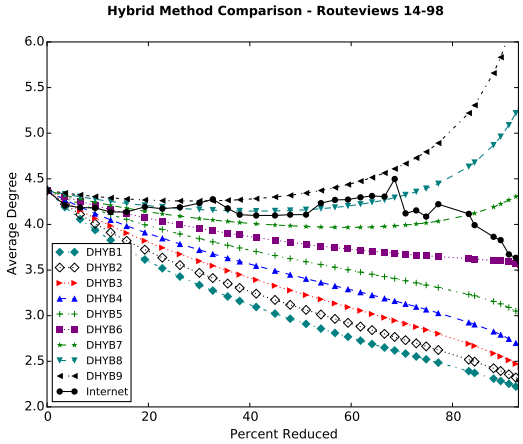

Figure 14: Average degree performance of DHYB methods - Routeviews 2014-1998

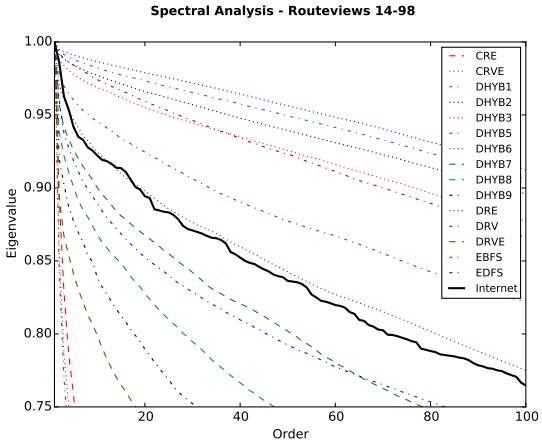

Figure 15: Spectra of reduction methods versus 1 January 1998 Internet graph - Routeviews 2014-1998

3.1 Routeviews Data Set: 2001-1998

As alluded to in Section 1, we first aimed to replicate the results of the authors of the original graph reduction experiments in [6]. We first compiled an inferred set of Internet instances, beginning with the 7 May 2001 instance, and ending with the instance observed on 24 January 1998. The 24 January 1998 graph serves as a target for each reduction to reach in terms of graph order. Further, because this graph is a known Internet instance, it also functions as a baseline to compare our reductions against. For intermediate points, we chose instances on the first of December, June, and March in between the start and end

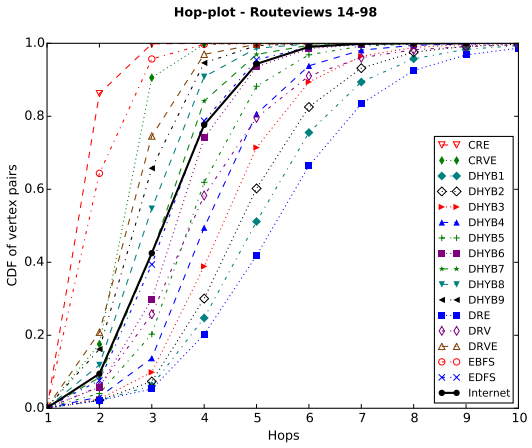

Figure 16: Hop-plot of reduction methods versus 1 January 1998 Internet graph - Routeviews 2014-1998

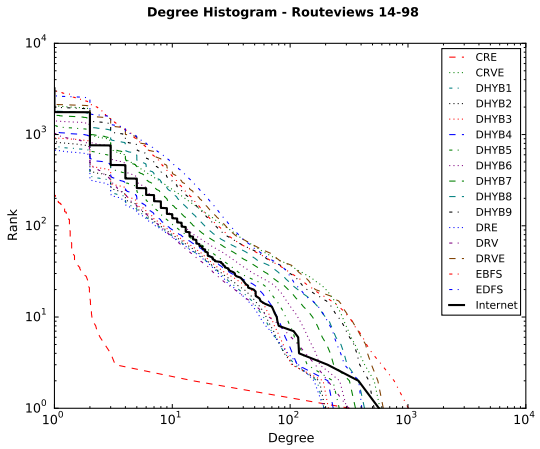

Figure 17: Degree histogram of reduction methods versus 1 January 1998 Internet graph - Routeviews 2014-1998

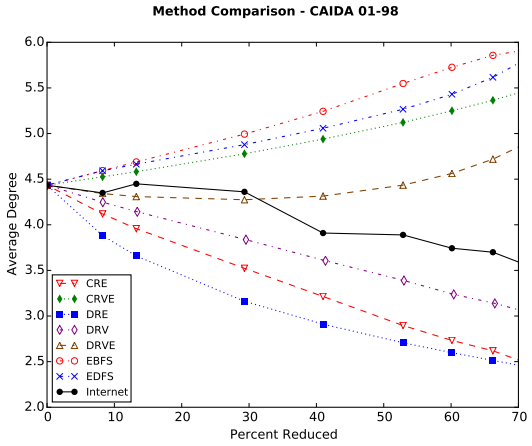

Figure 18: Average degree performance of non-DHYB methods - CAIDA 2001-1998

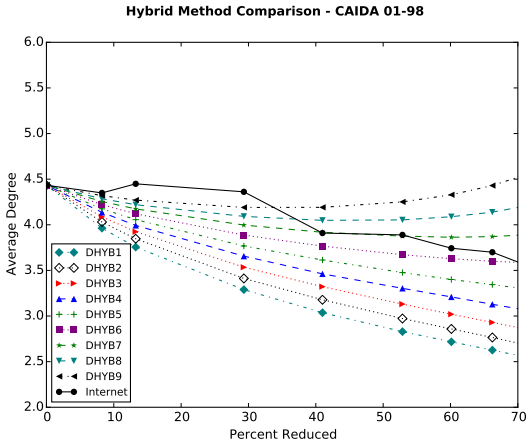

Figure 19: Average degree performance of DHYB methods - CAIDA 2001-1998

points when possible, or as close to the first as was available. When this data was compiled, a 25% difference in nodes between the initial May 2001 instance and the first intermediate point was noticed; for this reason, we also included points from January, February, March, and April 2001 for a more uniform sampling of inferred Internet data. For reference, the 7 May 2001 topology consists of 10,966 nodes and 22,536 edges, for an average degree of approximately 4.11, while the reduction end point of 24 January 1998 is a graph of order 3,291 and edge-set cardinality of 5,328, for an average degree of 3.52.

Using the 7 May 2001 inferred topology as our starting point, we proceed to reduce by all methods described in Section 2, taking averages of 50 different random seeds in all metrics

examined. When our reductions have reached a graph of order equivalent to one of our intermediate Internet instances, we capture this graph before continuing on to reduce it to a graph of approximately 3,291 nodes.

We first examine the average degree of the reductions over time to ensure consistency with prior work. In Figure 8, we observe similar results to that of [6]. None of the non-DHYB methods performs particularly well at the end reduction point, though DRVE follows the Internet line somewhat closely through 50% reduction in nodes.

When we examine the performance of the DHYB methods on this data set, we observe our first divergence from the conclusions drawn by the authors of [6]. While they find that the performance of DHYB-8 most closely matches the Internet's average degree at 24 January 1998 instance, we find that DHYB-7 performs even better. The DHYB methods plotted with the Internet's average degree line is shown in Figure 9.

Next, we turn to an analysis of the spectra of the most reduced Internet instance, and compare this to the average of the spectral behavior of our reduction methods of equivalent order. As in [6], we examine the 100 largest eigenvalues of the normalized adjacency matrix of these graphs. These spectra provide an insight into the amount of clustering that occurs within these networks, as described in [8].

In Figure 10, we see that our average of DHYB-8 reduced graphs performs well through eigenvalues of order 25-75, with some divergence from the Internet's spectra among eigenvalues of order 5-25 and the higher ordered eigenvalues above 75. While finding DHYB-8 to be the best reduction method in terms of spectral behavior matches the findings of [6], we see several anomalies between our analysis and theirs. First, a cursory look at the data presented in the prior work shows our reduction methods' plots to be far smoother than those in [6]. Second, prior work shows DHYB-8's reduction eigenvalues to be consistently higher than the Internet's, while our work shows them to be frequently lower. Based on an analysis of individual data points of our work, we believe that the authors of [6] have hand picked a particular reduction seed, rather than plotting the average all of the reduced graphs for a particular method.

Our next performance evaluation is *hop-plot*, which is defined as the CDF of reachable pairs of nodes in a graph within k or less "hops" along a geodesic. As with spectral analysis, we look only at the average of the reduction methods at the most reduced point and compare to the Internet graph's earliest point, 24 January 1998.

Several methods perform well in the *hop-plot* metric, as shown in Figure 11. DHYB-7 performs best at $k = 3$; differing by only .5% from the Internet's *hop-plot* line. At $k = 4$, we find that DHYB-6 offers the best performance, with 75.7% of reachable node-pairs compared to the Internet instance's 77.7%. DHYB-7 is again the best reduction method at

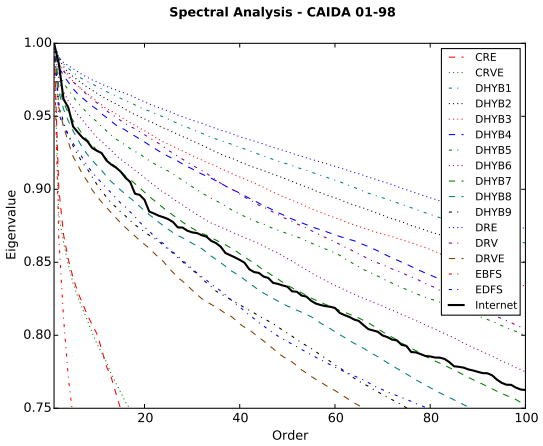

Figure 20: Spectra of reduction methods versus 1 January 1998 Internet graph - CAIDA 2001-1998

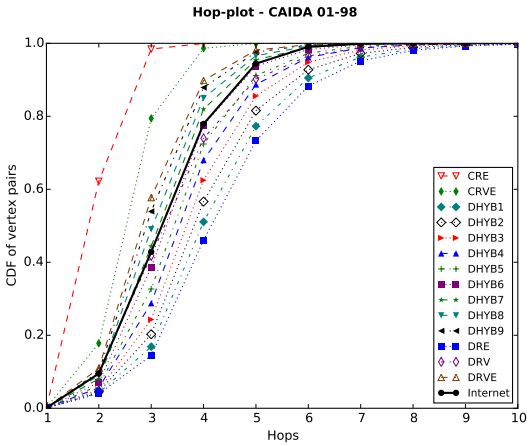

Figure 21: Hop plot of reduction methods versus 1 January 1998 Internet graph - CAIDA 2001-1998

$k = 5$, at which point 94.6% of node-pairs are reachable contrasted with the Internet graph's 94.8%. Prior work states that DHYB-8 is the best reduction method in terms of *hop-plot*.

Finally, we examine the degree distribution of the Internet inferred on 24 January 1998 as compared to the average degree distributions of the reduced graphs. This data appears in Figure 12.

While it is difficult to declare an absolute "winner" in terms of replicating the degree distribution of the Internet, we note here that several methods perform well at various points. First, CRVE comes the closest to matching the degree of the highest degree vertex of the Internet plot. It performs poorly throughout the rest of the vertices, however, producing consistently higher degrees than the Internet plot. Of the other methods, DHYB-5,6,7 all match the Internet degree distribution closely through the 10-100-degree vertex range. Because this metric was not studied in [6], we have no baseline to compare our results to.

3.2 Routeviews Data Set: 2014-1998

Our original work begins with considering Routeviews BGP RIB data over a longer period of time. In this data set, our most recent Internet instance is from 1 December 2014 - an inferred AS-level graph of order 49,185 and edge-set cardinality 107,517, for an average degree of 4.37. Our reduction end point is the order of the Internet instance inferred on 1 January 1998, and intermediate instances are evenly distributed on the first of June and first of December for all years available. Our reduction end point is a graph consisting of 3,211 vertices and 5,611 edges, which is a node reduction of approximately 93.5%. We consider the same metrics over this time period as we did in our reproduction of the original work.

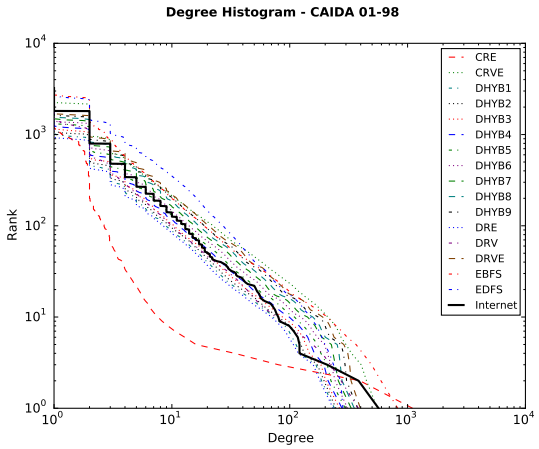

Figure 22: Degree histogram of reduction methods versus 1 January 1998 Internet graph - CAIDA 2001-1998

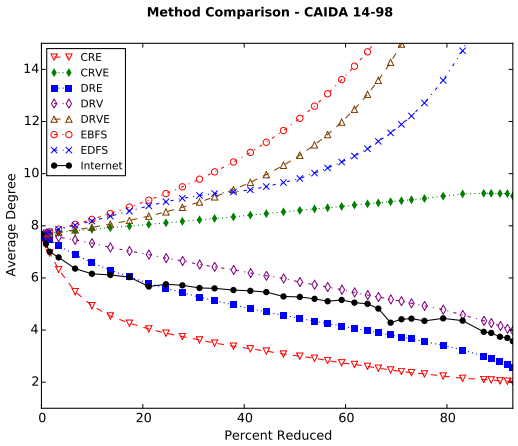

Figure 23: Average degree performance of non-DHYB methods - CAIDA 2014-1998

In this second data set, plotted in 13, we again see that none of the non-DHYB methods follow the Internet's average degree plot well, and that none come close to arriving at the most reduced instance's average degree of approximately 3.6. The best performing non-DHYB method, DRV, has an average degree difference of over 1 from the Internet at the most reduced point.

Of the DHYB methods, we see the best performance at the reduction end point from DHYB-6. The average of the DHYB-6 reduced average degrees is 3.56, which aligns closely with the average degree of the 1 January 1998 instance, 3.49. None of the curves is a good fit overall, however. DHYB-7, our best performer in Section 3.1, differs in average degree by approximately .84 from the Internet at our end point. The results of the DHYB reduction methods are shown in Figure 14.

Turning now to spectral behavior, we observe again that DHYB-6 outperforms all metrics in replicating the Internet's spectra. The spectral plot for this data set is seen in Figure 15.

Next, we compare the *hop-plot* performance of the reduction methods. While, as in section 3.1, we see that both DHYB-6 and DHYB do well, when $k = 4$, we observe the performance of EDFS most closely matching the percentage of reachable pairs within 4 hops in the Internet instance. Figure 16 displays our results.

Lastly, for the more current Routeviews data set, we again examine the results of the degree distributions produced by the various graph reduction methods in Figure 17. Of interest in this plot is that again CRVE performs well only at the highest degree vertex; similar to our results in Section 3.1, we again see that several DHYB reductions perform well throughout

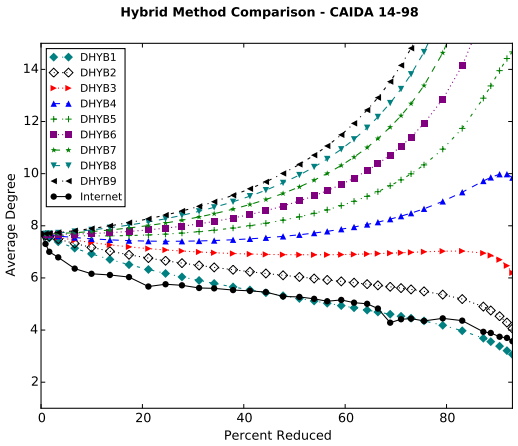

Figure 24: Average degree performance of DHYB methods - CAIDA 2014-1998

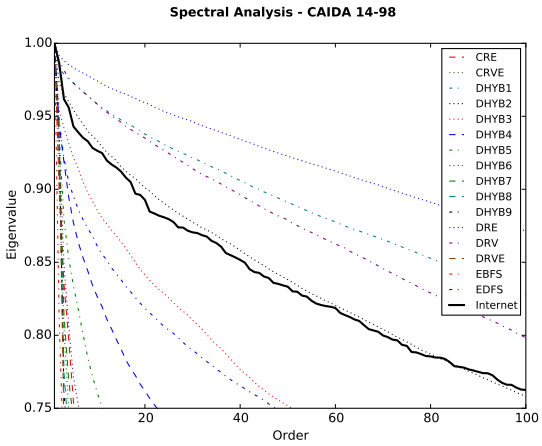

Figure 25: Spectra of reduction methods versus 1 January 1998 Internet graph - CAIDA 2014-1998

the distribution of vertices between 10 and 100. In this case, DHYB-4,5 and 6 most closely follow the Internet degree distribution.

3.3 CAIDA Data Set:2001-1998

In this section, we turn our attention to the data collected by CAIDA over the same time frame as the initial study done by the authors of [6]. Our initial instance comes from 1 May 2001, the closest data point compiled by CAIDA to the original analysis in Section 3.1. This graph consists of 11,045 nodes and 24,485 edges, for an average degree of approximately 4.43. It should be noted that while the order of the graph is similar to that obtained from the Routeviews data, the edge-set cardinality of the graph is larger by about 2000 edges. The reduction end point for this data set is the inferred Internet instance from 1 January 1998. The reduction end point graph contains 3,233 nodes and 5,773 edges, with an average degree of about 3.57. As in the prior data sets, we chose intermediate data points from the first of June and December between the start and end of our reduction time frame if possible; if not, an inferred graph close to those dates was selected.

Once again, our average degree experiments show that none of the non-DHYB reduction methods achieve the desired 3.57 of the reduction end point from January 1998. As with prior data, we note that DRVE tracks the Internet plot fairly well through 30% reduction, however. See Figure 18.

Of the DHYB reduction methods, we find that DHYB-6 performs the best at the most reduced point. Again, none of the curves fit the Internet's plot throughout the whole time period considered, however. Figure 19 displays these findings. This is similar to the original Routeviews data set in Section 3.1, in which we found DHYB-7 to be the winner of

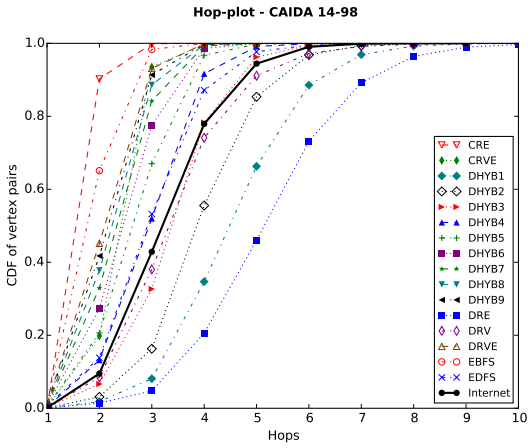

Figure 26: Hop-plot of reduction methods versus 1 January 1998 Internet graph - CAIDA 2014-1998

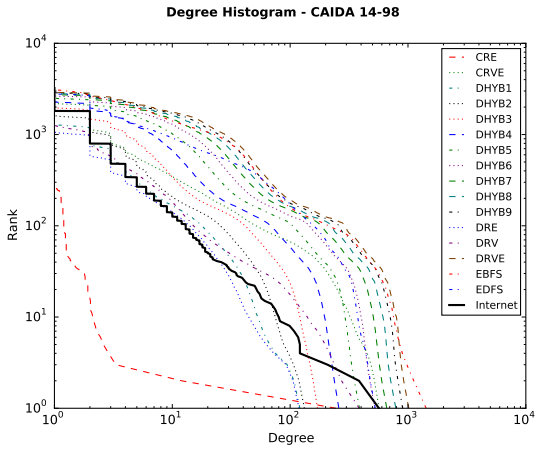

Figure 27: Degree histogram of reduction methods versus 1 January 1998 Internet graph - CAIDA 2014-1998

the average degree comparison, and the same as the extended Routeviews data set in Section 3.2.

When evaluating spectral behavior, DHYB-7 is the clear winner. This reduction method tracks the sorted eigenvalues extremely well throughout the majority of the top 100 considered. This is the first data set in which DHYB-7 has performed the best in this metric, but it is between the DHYB-8 and DHYB-6 found in Sections 3.1 and 3.2, respectively. Our data is displayed in Figure 20.

Our analysis of *hop-plot* data shows several reduction methods track with the Internet data at various $k-$hop distances. At $k = 2$, DRV, DHYB-8 and DHYB-9 all fall within 6% of the Internet's data point of 0.095. As k increases, DHYB-6 and DHYB-7 bound the Internet line, with DHYB-6 offering marginally better performance. This can be seen in Figure 21.

Finally, we examine the Internet's degree distribution at the 1 January 1998 reduction end point versus the averaged degree distributions of our reduced graphs. For the third time, CRVE most closely matches the highest degree node in the Internet instance, though it tracks the rest of the Internet line poorly. Further, as in Section 3.4, we find that DHYB-4,5 and 6 track the Internet's degree distribution most closely through the vertices of degree 10-100. Our results are displayed in Figure 22.

3.4 CAIDA Data Set: 2014-1998

Lastly, we examine the CAIDA data set from 1 December 2014 to 1 January 1998. Our largest Internet instance is a graph comprised of 46,177 nodes, with 177,391 edges for an average degree of 7.68. This average degree is the first major difference between this data

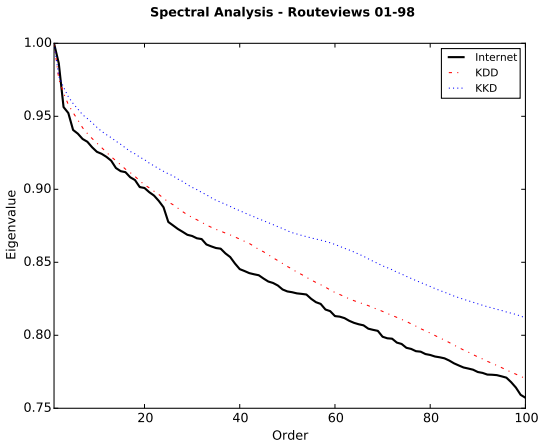

Figure 28: Spectral analysis of KKD and KDD - Routeviews 2001-1998

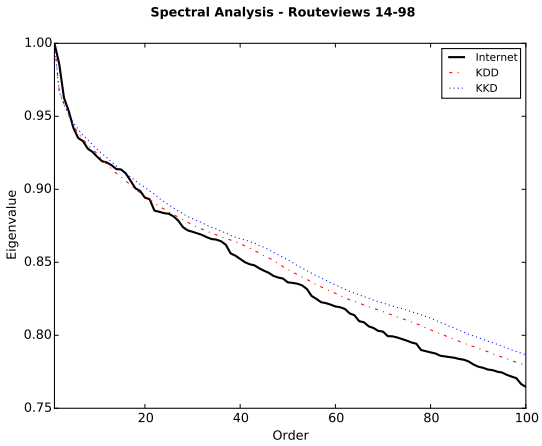

Figure 29: Spectral analysis of KKD and KDD - Routeviews 2014-1998

set and those studied earlier in this section; it is far higher than the roughly 3.5 of the other data sets. This graph is then reduced, using our reduction methods, to graphs that have a comparable number of nodes to the Internet instance from 1 January 1998, which has order 3,233 and edge-set cardinality of 5,773. This is the same reduction end point as Section 3.3.

In our examination of average degree performance of non-DHYB methods, we see that DRV comes within about 10% of the average degree of the most reduced Internet instance - DRV with 3.97, and the final Internet instance with 3.57. Our non-DHYB data is shown in Figure 23.

In the DHYB methods, we see for the first time very different results. DHYB-1 and DHYB-2 are the clear best reduction methods, and bound the January 1998 instance. With average degrees of 3.06 and 4.06, respectively, DHYB-1 and DHYB-2 do not match the end point's average degree as well as DRV. See Figure 24.

In our spectral analysis, plotted in Figure 25, the graphs reduced by the DHYB-2 reduction method exhibit the most similarity to the January 1998 Internet instance. Though the curve is slightly higher than that of the Internet for most of the 100 largest eigenvalues, the DHYB-2 slope easily bests all other methods. Further, in Figure 25, we note that all but four of the reduction methods (DHYB-1, DHYB-2, DRE, and DRV) produce spectral plots with normalized eigenvalues that quickly decrease below the .75 mark. This indicates that the amount of clustering is dramatically less than the Internet graph at the most reduced point for these reduction methods.

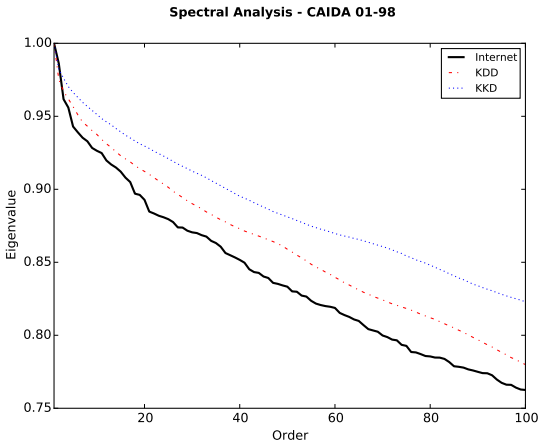

Figure 30: Spectral analysis of KKD and KDD - CAIDA 2001-1998

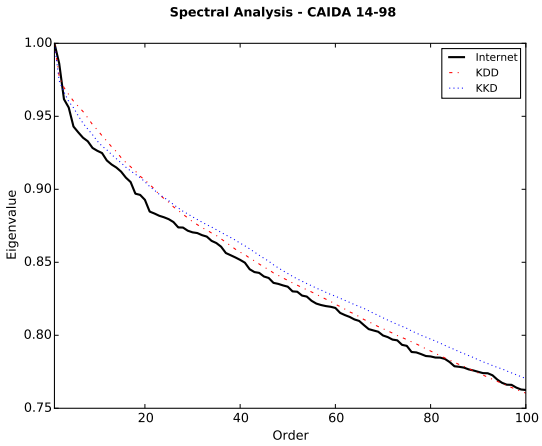

Figure 31: Spectral analysis of KKD and KDD-CAIDA 2014-1998

The *hop-plot* results for the extended CAIDA data set indicate interesting results as well, as seen in figure 26. DRV most closely matches the CDF of node-pairs reachable within k-hops through $k = 5$, with EDFS bounding the Internet curve above.

Lastly, we examine the degree distributions of our reduced graphs compared with the Internet instance end point. Again in Figure 27, we see results that significantly differ from those seen in previous sections. No method performs particularly well; CRVE again best matches the highest degree vertex, and DRE and DRV follow the distribution of vertices of degree 30 and less fairly closely, but no method can realistically be called effective.

3.5 k-core reductions

Based on prior work demonstrating that the k-core of Internet graphs exhibits self-similarity, we attempt to construct our own reduction methods. [9] Our goal is to produce a method that effectively preserves at least one metric across both data sets and time periods. To that end, we have produced two candidates we call KKD and KDD. Both methods require not only a target number of nodes to reach at the end of the reduction, but also a target number of edges. This is both a benefit and a disadvantage; in specifying a target number of edges to reach, we ensure that the average degree of the most reduced graph matches that of our reduction end point. It is disadvantageous in that we now require a second parameter at all - were we attempting to create Internet-like graphs for simulation or emulation purposes that were smaller than any available Internet instance, we would be providing at best an educated guess for edge-set cardinality.

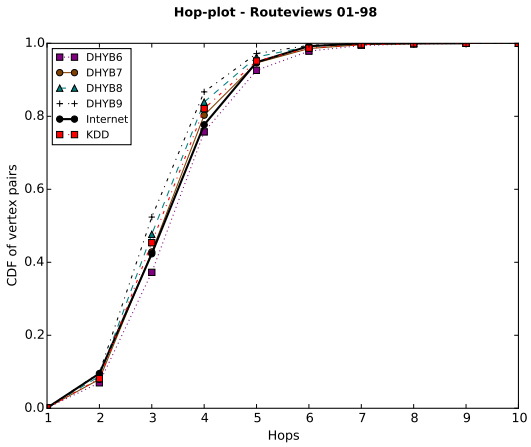

Figure 32: Hop-plot of KKD and KDD - Routeviews 2001-1998

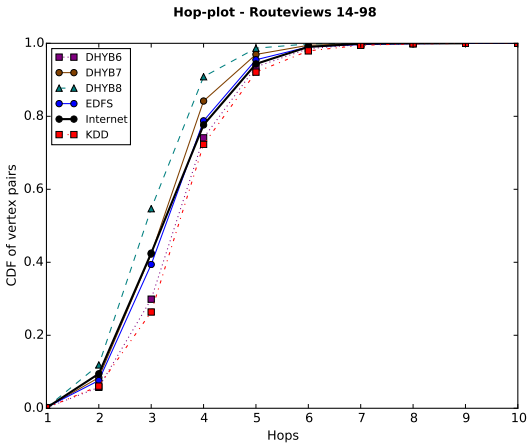

Figure 33: Hop-plot of KKD and KDD - Routeviews 2014-1998

3.5.1 KKD

In KKD, we begin with our initial Internet instance and take successive k-cores of the graph. This phase ends when the $(k+1)^{th}$-core would result in a graph of order smaller than our target, and the k-core of the original instance proceeds to the next phase. Here, we proceed by randomly removing vertices from the graph obtained from k-core process by selecting for removal vertices whose degree is exactly k. This is continued until the target node number is reached. Finally, the last stage consists of removing the excess edges to reach the target edge number. In order to preserve the target number of nodes we reached in the second stage, randomly selected edges whose removal does not disconnect the graph are removed. This is repeated until our target edge-set cardinality is reached, at which point our reduction is completed.

3.5.2 KDD

In KDD, we initially reduce the starting instance by taking repeated k-cores, as in KKD. When taking a $(k+1)^{th}$-core would produce a graph of order smaller than the target number of nodes, we begin using DRVE as described in Section 2.2 until the target number of nodes is achieved. From this point, we reduce by selecting random edges whose removal does not partition the graph until the target edge number is reached.

3.6 k-core results

We first note that we have no average degree plots to present; this is because, as noted earlier, KDD and KKD require a target edge number. Requiring this parameter allows us to reach our target average degree precisely.

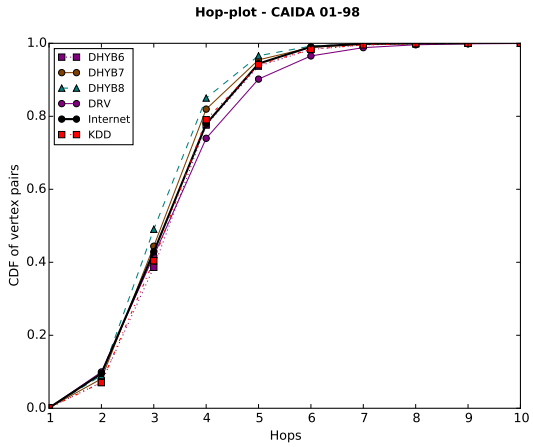

Figure 34: Hop-plot of KKD and KDD - CAIDA 2001-1998

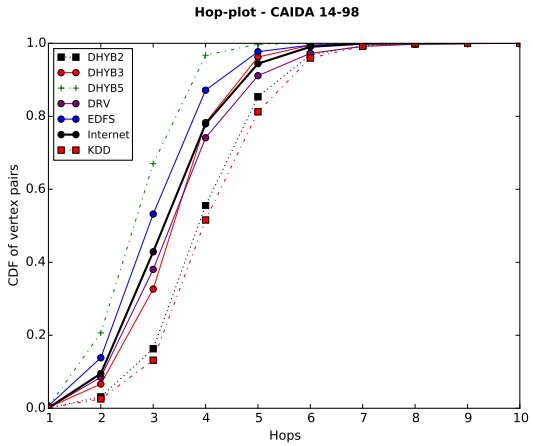

Figure 35: Hop-plot of KKD and KDD - CAIDA 2014-1998

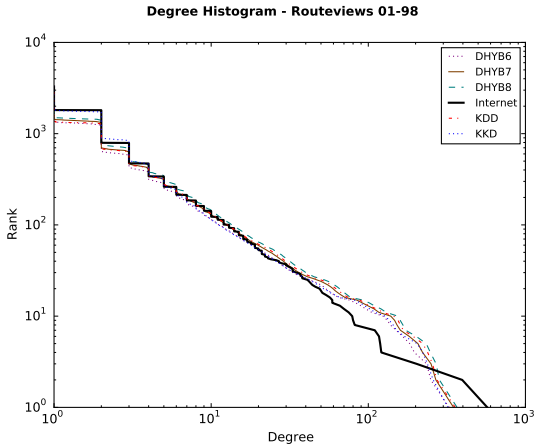

Figure 36: Degree histogram of KKD and KDD - Routeviews 2001-1998

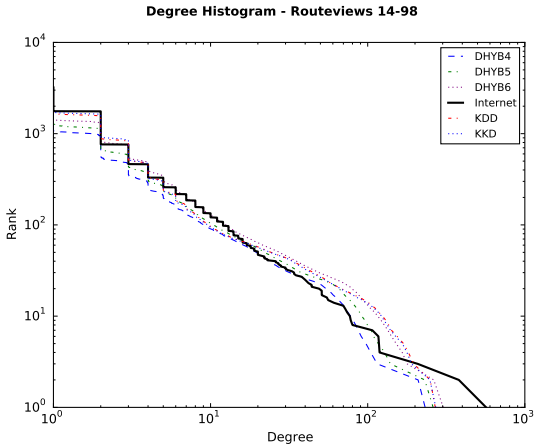

Figure 37: Degree histogram of KKD and KDD - Routeviews 2014-1998

Next, we present the plots of KDD and KKD reduced graph spectra against those of the reduction end points across both data sets and time periods. We note here that in both data sets, the longer time frame from 2014 to 1998 exhibits better performance than in the shorter time frame form 2001 to 1998. This is likely the case because more k-cores could be taken in the data sets over the larger time period - in both the CAIDA and Routeviews data sets beginning in 2014, we stopped our k-core reduction process at $k = 8$; in the data sets beginning in 2001, however, we had to stop at $k = 2$, and reduce much further using the k-deletion or DRVE phases. The ability to obtain k-cores for higher k values causes the more central ASes to be separated from the lower degree, lower clustering periphery. The

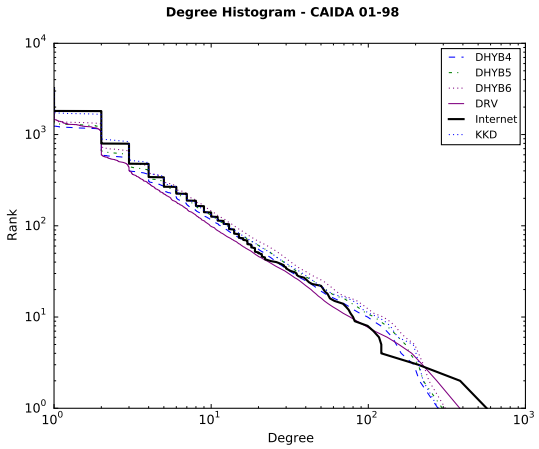

Figure 38: Degree histogram of KKD and KDD - CAIDA 2001-1998

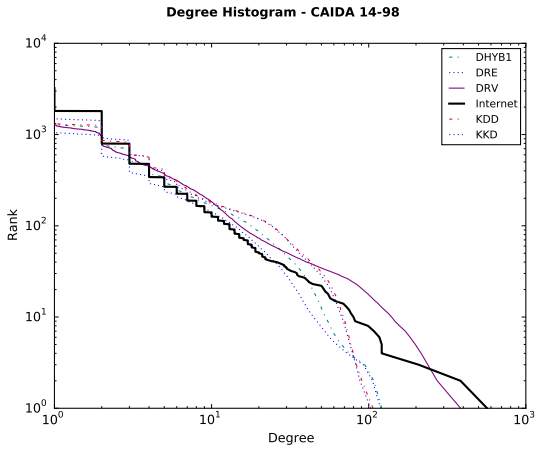

Figure 39: Degree histogram of KKD and KDD - CAIDA 2014-1998

difference in gaps between the Internet curve and the KKD and KDD curves in Figures 28 and 29 and Figures 30 and 31 demonstrate this phenomenon.

Further, we see the spectra of KDD outperform KKD in all four plots; the spectra of KKD are higher on average than those of both the Internet and KDD. We believe this is caused by the removal of only the k-degree vertices in the second step of KKD; in doing so, we tend to preserve the largest, most connected clusters within the graph.

Finally, we note that over the longer time frames, KDD performs nearly as well as the best performing DHYB reduction methods, despite the fact that these DHYB reduction methods differ in their probability value by 40%. We find this point, in particular, to be a good indicator that KDD is a valuable method for maintaining the spectral properties of the Internet graph, even through a high percentage of reduction.

In Figures 32, 33, 34, and 34, we show the *hop-plot* performance of KDD and KKD. Unlike our spectral results, we see above-average performance for KDD and KKD in the data sets from 2001-1998, while our plots do not follow the Internet curve well in the 2014-1998 period. In all plots, we note that KDD performs better than KKD.

Finally, in Figures 36, 37, 38, and 39, we show the degree histograms of KDD and KKD reduced graphs against the Internet reduction end point. In both data sets and time periods, our reduction methods produce highest-degree vertices of several hundred less than in the Internet instance. This is somewhat common, as we have seen previously. In three of these plots, Figures 36, 37, 38, we observe above-average performance from the 100-1 degree range from our reduction methods, KDD in particular. In Figure 39, however, both KKD and KDD reduction methods exhibit a tendency to create vertices of higher degree than

	RV 01-98	**RV 14-98**	**CAIDA 01-98**	**CAIDA 14-98**
Avg. Deg	DHYB-7	DHYB-6	DHYB-6	DHYB-1,2 DRV
Spectral	DHYB-8	DHYB-6	DHYB-7	DHYB-2
Hop Plot	DHYB-6,7	DHYB-6,7 EDFS	DHYB-6,7	DRV EDFS
Deg. Dist.	DHYB-5,6,7 CRVE	DHYB-4,5,6 CRVE	DHYB-4,5,6 CRVE	DRE, DRV CRVE

Table 1: Summary of best methods per data set and time frame considered

the Internet through the 20-110 rank range. Unlike the metrics examined previously, KKD follows the Internet curve slightly better than KDD throughout much of the degree range; KDD tends to produce a slightly higher degree high-degree vertex, however.

4 Conclusions

To begin, we first make a few remarks about the differences between our results from the Routeviews data set beginning in 2001 and the results found using the same data set by the authors of [6]. In the three metrics we studied that were also studied by the authors of the prior work, we found similar, but not always exactly the same best performing metric. Though the authors do not explicitly say so, their results indicate that they did not perform the reductions using all DHYB, one-tenth interval p values between 0 and 1. We make this claim because, in addition to not displaying results for values besides 1, 5, 6, and 8, they often claim that DHYB-8 is the "winner" of a particular metric, with DHYB-6 being a close second or runner-up. Our results show, as one would expect, that the DHYB p values produce an ordered performance; that is, if DHYB-x was the best hybrid method, then either DHYB-$(x+.1)$ or DHYB-$(x-.1)$ is necessarily the next best performing hybrid method.

Further, as mentioned in Section 3, we find the slope of their spectral plots to be indicative of one particular set of eigenvalues, rather than an average. This can be deceiving; often, these normalized eigenvalues can differ by more than 5% per ordered position. We believe that our methodology provides a more accurate description of expected behavior by using the arithmetic mean.

Turning back to our complete work, the results of the best performing metrics across the four time period and data set divisions we studied in Section 3 are summarized in Table 1.

We first make a few observations based on the data therein. First, the choice of data set and time period has a significant effect on the methods that will most closely replicate the

behavior of the Internet across the different structural metrics we evaluated. This is most obvious when comparing the results of the CAIDA 2014-1998 data sets with the other three data sets. The biggest difference between these data sets is that the average degree of the starting Internet instance for the CAIDA data from 2014 is nearly double that of the other three. Therefore, for any reduction method to be useful across data sets, it must mimic the behaviors of the end reduction point while being robust to changes of up to 100% in average degree.

Second, the CRVE reduction method, while a poor performer in terms of the other metrics, consistently reproduces the highest degree behavior of the Internet at the reduction end points.

Lastly, CRE performs poorly across all metrics. This is due to the likelihood of picking an edge attached to a high degree vertex in the graph in this method; high degree vertex clustering is quickly diminished, and hub nodes are likely to contract into each other, lowering the average path length beyond that of other reduction methods. The spectral analysis plots of Figures 10, 15, 20, and 25, and *hop-plot* Figures 11, 16, 21, and 26 confirm this hypothesis.

In terms of the performance of DHYB methods, we observe that across both data sets and time periods, in all metrics considered the Internet curve is bounded by DRE and DRVE. This indicates that there is likely a value of p for which DHYB-p offers the best fit for each of the structural characteristics considered. We hypothesize that an equation could be realized for which an exact p value could be calculated that would provide us with a DHYB-p to best match the metric in question.

Finally, our KDD and KKD reduction methods exhibit good spectral performance across data sets and time periods, with longer time periods providing a closer approximation to the Internet curve. This leads us to believe that targeting graph properties of the initial Internet instance will lead to more robust reduction methods that can handle dissimilarities like average degree, as observed in the extended CAIDA and Routeviews data sets. In particular, we believe that the k-core of the graph provides an avenue through which to target the most central nodes and important structure of the network, which can then be followed by reductions to the number of vertices and edges as appropriate for the reduction target.

References

[1] S. Knight, H. Nguyen, O. Maennel, I. Phillips, N. Falkner, R. Bush, and M. Roughan, "An automated system for emulated network experimentation," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 235–246.

[2] S.-H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the internet's large-scale topology," *Proceedings of the National Academy of Sciences*, vol. 99, no. 21, pp. 13 382–13 386, 2002.

[3] 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*. IEEE, 2001, pp. 346–353.

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

[5] J. P. Sterbenz, J. P. Rohrer, and E. K. Çetinkaya, "Multilayer network resilience analysis and experimentation on GENI," The University of Kansas, Lawrence, KS, ITTC Technical Report ITTC-FY2011-TR-61349-01, July 2010. [Online]. Available: http://www.ittc.ku.edu/resilinets/reports/Path_Diversity_Experiments_TR_public.pdf

[6] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J.-H. Cui, L. Lao, and A. G. Percus, "Sampling large internet topologies for simulation purposes," *Computer Networks*, vol. 51, no. 15, pp. 4284–4302, 2007.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *et al.*, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.

[8] C. Gkantsidis, M. Mihail, and E. Zegura, "Spectral analysis of internet topologies," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1. IEEE, 2003, pp. 364–374.

[9] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani, "k-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases," *arXiv preprint cs/0511007*, 2005.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California