Revisiting AS-Level Graph Reduction

The Eighth IEEE International Workshop on Network Science for Communication Networks

Erik C. Rye*, Justin P. Rohrer+, Robert Beverly+

*US Naval Academy
Annapolis, MD
+Naval Postgraduate School
Monterey, CA

11 April 2016

Outline

Motivation

- Long-standing need to model macroscopic behavior of the Internet
- e.g., at the Autonomous System (AS) level: ISPs as nodes and links as their (complex) interconnection
  - Evaluate new routing protocol
  - Understand provider filtering (BCP38, SBGP, etc)
  - Active topology mapping (our particular motivation)
- But. . .
  - Size of entire-Internet AS graph makes emulation infeasible and simulation difficult
  - Thus, a need for smaller, "representative" Internet models exists
  - But what is representative?
    - Degree distribution? Clustering? Avg. path len?
  - And how?
    - Constructive – build graph from ground-up
    - Reductive – begin with AS graph, pare down

Motivation

- Long-standing need to model macroscopic behavior of the Internet
- e.g., at the Autonomous System (AS) level: ISPs as nodes and links as their (complex) interconnection
  - Evaluate new routing protocol
  - Understand provider filtering (BCP38, SBGP, etc)
  - Active topology mapping (our particular motivation)
- But. . .
  - Size of entire-Internet AS graph makes emulation infeasible and simulation difficult
  - Thus, a need for smaller, "representative" Internet models exists
  - But what is representative?
    - Degree distribution? Clustering? Avg. path len?
  - And how?
    - Constructive – build graph from ground-up
    - Reductive – begin with AS graph, pare down

Our Contribution

- Re-evaluation of prior sampling (reductive) algorithm on multiple modern Internet graphs
- Development of new graph sampling algorithms that out perform existing techniques on modern Internet graphs

Prior Work

- Lots of prior work on Constructive Internet graph generators
- We focus on reduction:
  - Cem *et al.*– Induced Random Vertex, Random Walk, Random Edge sampling on varied networks
  - Vaquero *et al.*– Breadth-First Search to reduce backbone AS architecture for end-to-end delay estimation
  - Krishnamurthy, Faloutsos

Prior Work Continued

- Krishnamurthy, Faloutsos, *et al.*.
  - *Sampling large Internet topologies for simulation purposes*
  - Start with May 2001 AS-level graphs of the Internet
  - Data obtained passively, obtained from RouteViews Border Gateway Protocol (BGP) Router Information Base (RIB) dumps
  - Reduce these graphs using 16 different methodologies to target reduction order – Jan 1998 Internet instance
  - Compare fidelity of reduced graphs to Jan 1998 Internet graph metrics

Use their methodology as a starting point. . .

- . . . but draw from chronologically newer data
- . . . expand data sources
- . . . and improve with new algorithm

Methodology

We successfully replicate the results of Krishnamurthy *et al.*:

- Contraction
  - Contract two endpoints of an edge together into new node
  - New node retains all edges incident to original two nodes
- Deletion
  - Delete randomly selected node or edge
  - How we pick edges, in particular, affects resultant topology
- Exploration
  - Use Breadth/Depth - First Search strategies

We consider the same methods, and introduce two novel sampling strategies based on the graph's k-core.

k-core Reduction

Our approach

- Prior work shows k-cores of the AS-level Internet graph exhibits self-similarity to complete AS-level Internet graph (Alvarez-Hamelin *et al.*, Zhou *et al.*)
- Implement reduction by computing successive k-cores (until $(k+1)^{st}$-core contains too few vertices), then either:
  - **KDD:** reduce by removing edge incident to random vertex, then delete random edges.
  - **KKD:** or reduce by removing nodes with degree k to meet vertex count, then delete random edges.

Methodology Continued

- We compare the reduced graphs (of the Jan 1998 Internet order) to the actual Jan 1998 AS-level Internet graph in the following metrics:
  - Average degree
  - Clustering, using the 100 largest eigenvalues of normalized adjacency matrix (normalized graph spectra)
  - Hop-plot (% of vertex pairs reachable within x hops along a geodesic)
  - Degree distribution
- First three metrics studied in prior work; degree dist. added for more fine-grained degree comparison

Data Sets

Dataset	Source	Construction	Time Frame
RV1	RouteViews	Observed AS_PATH	01/1998 - 05/2001
RV2	RouteViews	Observed AS_PATH	01/1998 - 12/2014
CAIDA1	CAIDA ITDK	Traceroute	01/1998 - 05/2001
CAIDA2	CAIDA ITDK	Traceroute	01/1998 - 12/2014

Summary of Results

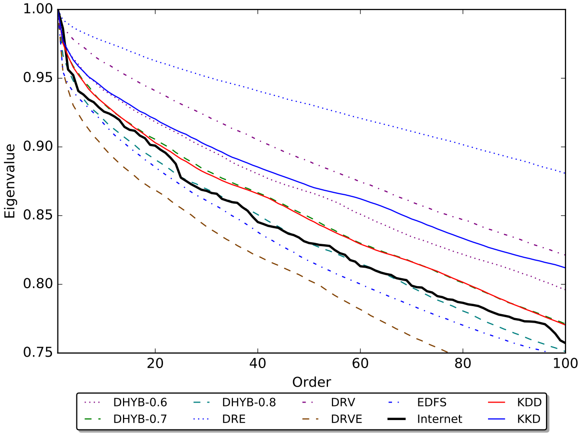
	RV1	RV2	CAIDA1	CAIDA2
Avg. Deg	DHYB-0.7	DHYB-0.6	DHYB-0.6	DHYB-0.1
Spectral	DHYB-0.8 **KDD** DHYB-0.7 EDFS	DHYB-0.6 **KDD** **KKD** DHYB-0.7	DHYB-0.7 DHYB-0.8 DHYB-0.6 **KDD**	DHYB-0.2 **KDD** **KKD** DHYB-0.1
Hop Plot	DHYB-0.7 **KDD** DHYB-0.8 DHYB-0.6	EDFS DHYB-0.7 DHYB-0.6 **KDD**	**KDD** DHYB-0.7 DHYB-0.6 DRV	DHYB-0.3 DRV EDFS DHYB-0.4
Deg. Dist.	**KKD** DHYB-0.7 DHYB-0.6 **KDD**	**KKD** **KDD** DHYB-0.5 DHYB-0.6	**KKD** DHYB-0.5 DHYB-0.4 DHYB-0.6	DHYB-0.1 DRE DRV **KKD**

Summary of Results

	RV1	**RV2**	**CAIDA1**	**CAIDA2**
Avg. Deg	DHYB-0.7	DHYB-0.6	DHYB-[illegible]	[illegible]
Spectral	DHYB-0.8	DHYB-0.6	DHYB-[illegible]	[illegible]
	KDD	**KDD**	DHYB-[illegible]	[illegible]
	DHYB-0.7	**KKD**	DHYB-[illegible]	[illegible]
	EDFS	DHYB-0.7	**KDD**	[illegible]
Hop Plot	DHYB-0.7	EDFS	**KDD**	[illegible]
	KDD	DHYB-0.7	DHYB-[illegible]	[illegible]
	DHYB-0.8	DHYB-0.6	DHYB-[illegible]	[illegible]
	DHYB-0.6	**KDD**	DRV	[illegible]
Deg. Dist.	**KKD**	**KKD**	**KKD**	[illegible]
	DHYB-0.7	**KDD**	DHYB-[illegible]	[illegible]
	DHYB-0.6	DHYB-0.5	DHYB-[illegible]	[illegible]
	KDD	DHYB-0.6	DHYB-[illegible]	[illegible]

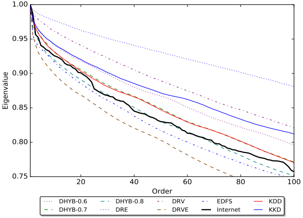
- By construction, KDD and KKD match avg. degree exactly
- While DHYB does well, it is sensitive to parameterization
- Our algorithms perform well w/o parameters

Reduced Graph Spectra

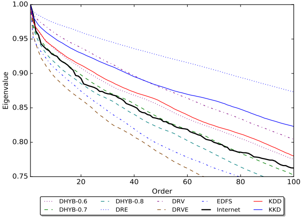


- (See paper for full metrics comparison)
- Spectra of **KDD** closely matches target Internet instance
- What about other time periods and data sources?

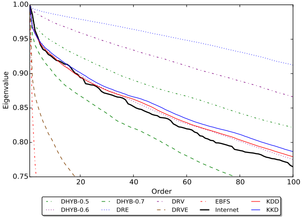
Reduced Graph Spectra



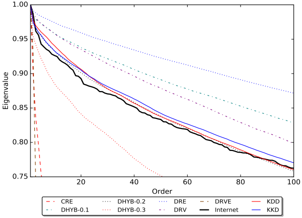
RV1



CAIDA1



RV2



CAIDA2

Conclusions

- Previous best reduction methods differ considerably across time periods and AS-graph inference methods
  - DHYB often a good choice, but probability values fluctuate wildly
- Leveraging Internet AS graph properties more promising than random deletion methods
  - *k*-core-based reduction algorithms consistently in top 4 reduction methods across data sources and time frames
  - *k*-core reduction methods match average degree of target graph precisely
- Our implementation is publicly available at https://github.com/cmand/graphreduce

Thanks! Questions?