A Multi-Level Approach for Evaluating Internet Topology Generators

Ryan Rossi[1], **Sonia Fahmy**[1], **Nilothpal Talukder**[1,2]
[1]Purdue University, IN
[2]Rensselaer Polytechnic Institute, NY
Email: {rrossi,fahmy}@cs.purdue.edu, talukn@cs.rpi.edu

May 23, 2013

Motivation

Why Topology Generators?

- Generate representative network topologies of different sizes
- Used for experiments to design protocols, predict performance, and understand robustness and scalability of the future Internet
- Unfortunately, many fail to capture static and evolutionary properties of today's Internet, e.g., assume average path length and clustering coefficient constant

Our goal:

- Determine how to quantitatively assess a generator through a multi-level hierarchy of graph, node and link measures
- Focus on 2 popular generators: **Orbis** [SIGCOMM06] and **WIT** [INFOCOM07]
- Validate using different views of the Internet: *data* (traceroute), *control* (BGP tables), and *management* (WHOIS) planes

Taxonomy of Topology Generators

Generators	Process	Model Type	Topology
WIT	**Random-walks**	Parametric	AS
RSurfer		Parametric	N/A
Orbis	**Optimization**	Data-driven	AS & RL
HOT		Parametric	RL
Mod. HOT		Parametric	AS
AB	**Preferential Attachment**	Parametric	N/A
BRITE		Data-driven	AS & RL
Inet		Parametric	AS
GLP		Parametric	AS
SWT	**Geometry**	Parametric	AS & RL
GT-ITM		Parametric	AS & RL

Orbis Topology Generator [SIGCOMM 2006]

- Series of measures based on degree correlations
- The first few dK distributions are:
  - $0K$ (average degree)
  - $1K$ (degree distribution: $P(k) = n(k)/n$)
  - $2K$ (joint degree distribution: $P(k_1, k_2) = m(k_1, k_2)\mu(k_1, k_2)/(2m)$, where $\mu(k_1, k_2) = 2$ if $k_1 = k_2$, otherwise 1)
  - $3K$ (wedges and triangles), etc.
- Fails to capture global characteristics
- d must be small in practice due to increasing complexity
- Relies on rescaling technique; inaccurate as topology becomes larger

WIT Topology Generator [INFOCOM 2007]

- Captures the "wealth" of ISPs over time
- Multiplicative stochastic process, $u_i(t) = \lambda_i(t)\, u_i(t-1)$, where $u_i(t)$ is the unscaled wealth and $\lambda_i(t)$ is an independent random variable
- $w_i(t)$ is the normalized wealth for node i, and $z_i(t) = C \cdot d_i(t)$ is the expense
- In each iteration,
  - If $w_i(t) - z_i(t) > C + T$, place a link between the node i and an arbitrary node by randomly walking l-steps from i
  - If $w_i(t) - z_i(t) > -T$, remove a random link of node i
- Threshold T is carefully chosen to avoid oscillation

Orbis versus WIT

- WIT attempts to model the evolution of the AS topology
  - Fails when the underlying process and growth of the Internet change
- Orbis generates topologies that preserve a set of measures
  - Fails if the set of characteristics is incomplete w.r.t. the actual AS topology
- What is the best representative set of local and global measures?

Network Properties

Measure		Importance in Computer Networks
LOCAL	Degree	Fault tolerance, local robustness
	Assortativity	
	Clustering coefficient	Path diversity, fault tolerance, local robustness
GLOBAL	Distance	Scalability, performance, protocol design
	Betweenness	Traffic engineering, potential congestion points
	Eigenvector	Network robustness, performance, clusters/hierarchy, traffic engineering

Measures Used

The order of evaluation measures in terms of the difficulty of preservation

Link $\geq$ Node $\geq$ Graph

Graph Measures

- Traditional: Average degree, Assortativity coefficient, Average clustering and Average distance, etc.
- Additional: largest singular value (λ_1), Network conductance ($\lambda_1 - \lambda_2$), radius, and diameter, etc.

Node Measures

- Traditional: Degree distribution, Clustering coefficient, distance, eccentricity, betweenness, etc.
- Additional: Network values, Scree Plots, K-walks, K-core, etc.

Measures Used (cont'd)

Link Measures

- Order of the nodes with respect to the magnitude of their coordinates along the principal direction
- The closest k-approximation of the topology

Community measures

- Louvain's modularity

Measures Used (cont'd)

Quantitative Measures

- Graph based:
  - The normalized root-mean-square error (NRMSE)

$$D_{NRMSE}(\vec{x}, \hat{\vec{x}}) = \frac{\mathbb{E}[(\vec{x} - \hat{\vec{x}})^2]}{\max(\vec{x}, \hat{\vec{x}}) - \min(\vec{x}, \hat{\vec{x}})}.$$

- Node based:
  - Kolmogorov-Smirnov (KS): $KS(F_1, F_2) = \max_x |F_1(x) - F_2(x)|$.
  - Kullback-Leibler (KL) divergence:

$$D_{KL}(P||Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}.$$

Learning Graph Measures [ReFeX, SIGKDD 2011]

Instead of selecting a set of graph measures, we automatically learn a set of graph measures recursively.

1. **Base set of measures.** The process starts by computing degree (in/out/total edges) and egonet measures (in/out egonet).
   - egonet includes the node, its neighbors, and any edges in the induced subgraph on these nodes.
2. **Aggregate measures.** The existing measures of a node are aggregated to create additional measures by taking the sum/mean of the neighbors (done in a recursive fashion). One simple measure is the mean value of the degree among all neighbors of a node.
3. **Prune correlated measures.** At each iteration, we test for redundant measures using a simple correlation test, and remove all measures that are highly correlated.
4. **Stopping Criteria.** Repeat steps 2-3 until no new measures are retained.

Evaluation Strategy

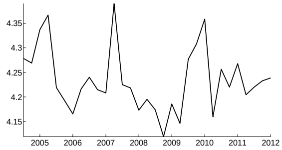
1. Given $G_n^\star$ of size n, generate same size graph G_n s.t. $\mathcal{M}(G_n) \approx \mathcal{M}(G_n^\star)$
2. Given $G_n^\star$ of size n, generate G_m of size m where $m \geq n$ s.t. $\mathcal{M}(G_m) \approx \mathcal{M}(G_n^\star)$
3. Given an ordered sequence $G_t^\star$ for $t = 1, 2, ..., m$, generate a corresponding sequence G_t for $t = 1, 2, ..., m$ s.t. G_t is the same size as $G_t^\star$ and $\mathcal{M}(G_t) \approx \mathcal{M}(G_t^\star)$

Datasets for Validation

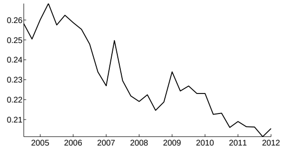
- Skitter traceroute
- RouteViews' BGP tables (RV)[1]
- RIPE's WHOIS
- HOT
- RocketFuel

[1] AS-level subgraphs for 2004-2012

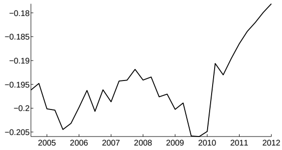
Results: Graph Measures



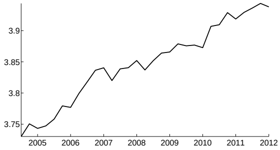
(a) Average Degree



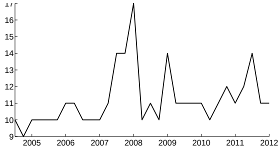
(b) Average Clustering



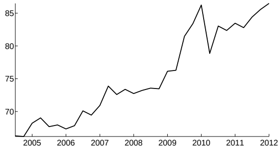
(c) Assortativity



(d) Characteristic Path Length

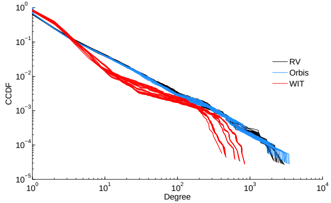


(e) Diameter

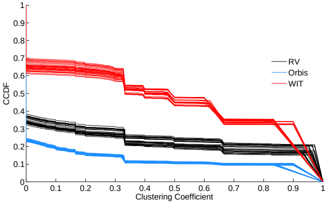


(f) Largest Eigenvalue

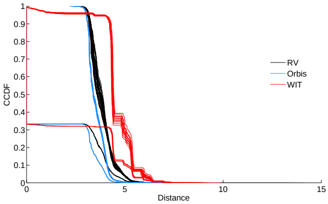
Results: Node Measures



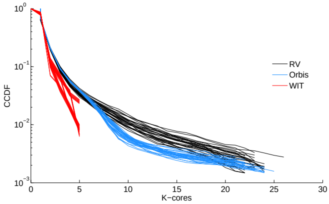
(a) Degree



(b) Clustering Coefficient

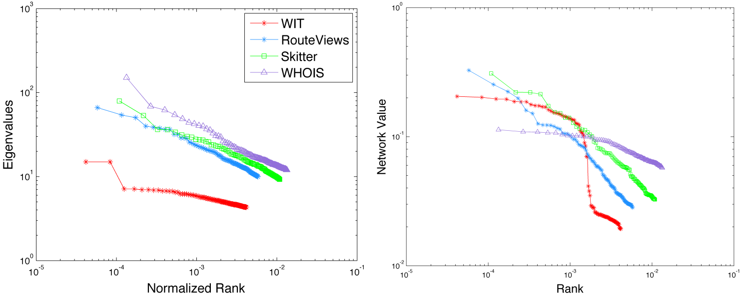


(c) Distance



(d) K-cores

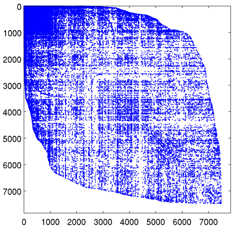
Results: Node Measures



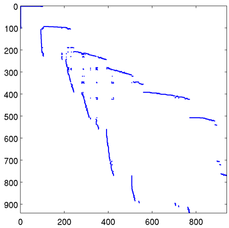
(a) Scree plot

(b) Network values

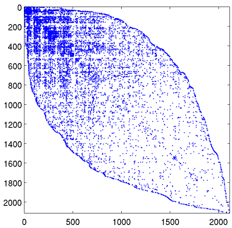
Results: Link Measures



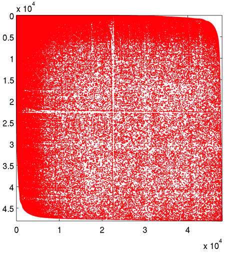
(a) WHOIS



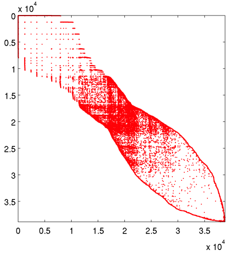
(b) HOT



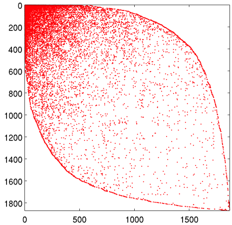
(c) RocketFuel



(d) Orbis (WHOIS)



(e) Orbis (HOT)



(f) Orbis (RocketFuel)

Results: Quantitative Measures

Table : Quantitative Evaluation of Orbis using KS Distance.

	Deg.	**CC**	**Ecc.**	**Kcores**	**PR**	**EigDiff**	**Net-Value**
HOT	0.009	0.000	0.000	0.078	0.067	**0.588**	0.131
RF	0.013	**0.450**	0.000	0.088	**0.215**	**0.629**	**0.680**
WHOIS	0.059	**0.480**	**0.224**	0.060	**0.536**	0.169	0.159
SKITTER	0.010	**0.211**	0.029	0.009	**0.342**	0.096	0.182

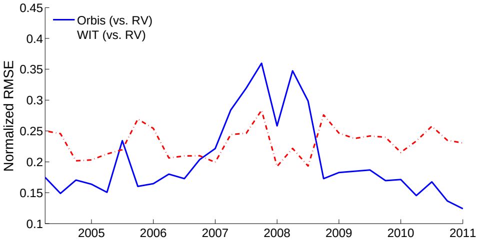
Results: Community Measures

Table : Evaluating the Community Structure of the Topologies.

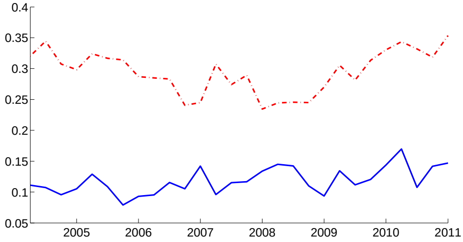
		Communities	Q	Nodes	Edges	Degree	CC
2004	RouteViews	*24*	*0.65*	3951	13360	3.38	0.45
	Orbis	*46*	*0.48*	957	2254	2.36	0.10
	WIT	*57*	*0.92*	755	2653	3.51	0.64
2011	RouteViews	*34*	*0.68*	6048	18496	3.06	0.22
	Orbis	*60*	*0.48*	2347	5640	2.40	0.12
	WIT	*66*	*0.94*	2095	11727	5.60	0.45

		Communities	C-path	Radius	Diameter
2004	RouteViews	*24*	2.74	3	6
	Orbis	*46*	3.01	4	8
	WIT	*57*	2.75	4	7
2011	RouteViews	*34*	3.27	5	9
	Orbis	*60*	2.91	4	8
	WIT	*66*	3.44	5	10

Results: Selected versus Learned Measures

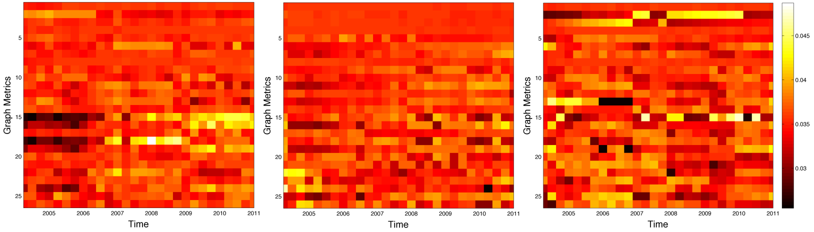


(a) Selected Measures



(b) Learned Measures

Results: Learned Graph Measures



(a) RV (Internet) (b) Orbis (c) WIT

Conclusions

- We propose a multi-level framework for understanding Internet topologies, and evaluating generators (focus on Orbis, WIT)
- We leverage both macro measures (graph) and micro measures (node and link measures) to accurately compare topologies
- We show that the existing generators fail to capture static and evolutionary properties of the Internet AS topology
- Data-driven generators generate static topologies with little or no variance
- Parametric generators typically cannot accurately model Internet evolution

Future Directions

- Investigate additional topology generators
- Develop a parameter estimation technique for WIT and analyze its behavior with the refined parameters
- Study Internet evolution and investigate causes for the changes we observed

Thank you.
Questions?