Degree correlations and topology generators

Dmitri Krioukov
dima@caida.org

Priya Mahadevan and Bradley Huffaker

5th CAIDA-WIDE Workshop
What’s the problem?

- Veracious topology generators. Why?
  - New *routing* and other protocol design, development, and testing
    - Scalability
      - For example: new routing might offer $X$-time smaller routing tables for today but scale $Y$-time worse, with $Y \gg X$
  - Network robustness, resilience under attack
  - Traffic engineering, capacity planning, network management
  - In general: “what if”
Veracious topology generators

- Reproducing closely as *many* topology characteristics as possible. Why “many”?  
  - Better stay on the safe side: you reproduced characteristic $X$ OK, but what if characteristic $Y$ turns out to be also important later on and you fail to capture it?  
  - Standard storyline in topology papers: all those before us could reproduce $X$, but we found they couldn’t reproduce $Y$. Look, we can do $Y$!

- Emphasis on practically *important* characteristics
Important topology characteristics

- Distance (shortest path length) distribution
  - Performance parameters of most modern routing algorithms depend solely on distance distribution
  - Prevalence of short distances makes routing hard (one of the fundamental causes of BGP scalability concerns (86% of AS pairs are at distance 3 or 4 AS hops))

- Betweenness distribution

- Spectrum
How to reproduce?

- Brute force doesn’t work
  - There is no way to produce graphs with a given form of any of important characteristics
  - Even more so for combinations of those

- More intelligent approach
  - What are the inter-dependencies between characteristics?
  - Can we, by reproducing most basic, simple, but not necessarily practically relevant characteristics, also reproduce (capture) all other characteristics, including practically important?
  - Is there the one(s) defining all other?

- We answer positively to these questions
Maximum entropy constructions

- Reproduce characteristic \( X (0K, 1K, \text{ etc.}) \)
  but make sure that the graph is *maximally random* in all other respects

- Direct analogy with physics (maximum entropy principle)
Most basic characteristics: Connectivity

<table>
<thead>
<tr>
<th>Tag</th>
<th>Name</th>
<th>Correlations of degrees of nodes at distance:</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0K</td>
<td>Average node degree</td>
<td>None</td>
<td>&lt;k&gt;</td>
</tr>
<tr>
<td>1K</td>
<td>Node degree distribution</td>
<td>0</td>
<td>(P(k))</td>
</tr>
<tr>
<td>2K</td>
<td>Joint node degree distribution or edge degree distribution</td>
<td>1</td>
<td>(P(k_1, k_2))</td>
</tr>
<tr>
<td>3K</td>
<td>Joint edge degree distribution</td>
<td>2</td>
<td>(P(k_1, k_2, k_3))</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>DK</td>
<td>Full degree distribution</td>
<td>D = maximum distance (diameter)</td>
<td>(P(k_1, k_2, ..., k_D))</td>
</tr>
</tbody>
</table>
Tells you
- Average node degree (connectivity) in the graph
  \[ <k> = \frac{2m}{n} \]

Maximum entropy construction (0K-random)
- Connect every pair of nodes with probability
  \[ p = \frac{<k>}{n} \]
- Classical Erdös-Rényi random graphs
- \[ P(k) \sim e^{-<k>} \frac{<k>^k}{k!} \]
Tells you

- Probability that a randomly selected node is of degree \( k \)
  \[ P(k) = \frac{n(k)}{n} \]
- Connectivity in 0-hop neighborhood of a node

Defines

- \(<k> = \sum_k k P(k)\)
Maximum entropy construction (1K-random)

1. Assign $n$ numbers $q$’s (expected degrees) distributed according to $P(k)$ to all the nodes;
2. Connect pairs of nodes of expected degrees $q_1$ and $q_2$ with probability
   
   \[ p(q_1,q_2) = \frac{q_1 q_2}{\langle n < q \rangle} \]

- More care to reproduce $P(k)$ exactly
- Power-law random graph (PLRG) generator
- Inet generator
Tells you

- Probability that a randomly selected edge connects nodes of degrees $k_1$ and $k_2$
  \[ P(k_1,k_2) = \frac{m(k_1,k_2)}{m} \]
- Probability that a randomly selected node of degree $k_1$ is connected to a node of degree $k_2$
  \[ P(k_2|k_1) = \langle k \rangle \frac{P(k_1,k_2)}{(k_1 P(k_1))} \]
- Connectivity in $l$-hop neighborhood of a node
2K

- Defines

- $\langle k \rangle = \left[ \Sigma_{k_1,k_2} P(k_1,k_2)/k_1 \right]^{-1}$

- $P(k) = \langle k \rangle \Sigma_{k_2} P(k,k_2) / k_2$
Maximum entropy construction ($2K$-random)

1. Assign $n$ numbers $q$’s (expected degrees) distributed according to $P(k)$ to all the nodes; 
2. Connect pairs of nodes of expected degrees $q_1$ and $q_2$ with probability 
   
   $p(q_1, q_2) = (<q> / n) P(q_1, q_2) / (P(q_1)P(q_2))$

- Much more care to reproduce $P(k_1, k_2)$ exactly
- Have not been studied in the networking community
3K

- Tells you
  - Probability that a randomly selected pair of edges connect nodes of degrees $k_1$, $k_2$, and $k_3$
  - Probability that a randomly selected triplet of nodes are of degrees $k_1$, $k_2$, and $k_3$
  - Connectivity in 2-hop neighborhood of a node

- Defines
  - $<k>$
  - $P(k)$
  - $P(k_1,k_2)$

- Maximum entropy construction ($3K$-random)
  - Unknown
What’s going on here?

- As $d$ increases in $dK$, we get:
  - More information about local structure of the topology
  - More accurate description of node neighborhood
  - Description of wider neighborhoods

- Analogy with Taylor series
  - Connection between spectral theory of graphs and Riemannian manifolds

- Conjecture: $DK$-random versions of a graph are all isomorphic to the original graph $\iff DK$ contains full information about the graph
Do we need to go all the way through to $DK$, or can we stop before at $d << D$?

- **Known fact #1**
  - $0K$ works bad

- **Known fact #2**
  - $1K$ works much better, but far from perfect in many respects

- Let’s try $2K$!
What we did

- Understood and formalized all this stuff
- Devised an algorithm to produce $2^K$-random graphs with exactly the same $2^K$ distribution
- Checked its accuracy on Internet AS-level topologies extracted from different data sources (skitter, BGP, WHOIS)
What worked

- All characteristics that we care about exhibited perfect match
Example: distance in BGP

![Graph showing distance distribution in BGP tables and random 2-k networks.](image-url)
Example: distance in skitter
What did not work

- Clustering
  - Expected to be captured by $3K$

- Router-level
  - Expected to be captured by $dK$, where $d$ is a characteristic distance between high-degree nodes
Main contribution

0K

1K

2K

3K

... 

DK
Future work

- Clustering in $3K$-random graphs
- Given a class of graphs, find $d$ such that $dK$-random graphs capture all you need
- Generalize maximum entropy construction algorithm for $dK$-random graphs with any $d$
More information

- “Comparative Analysis of the Internet AS-Level Topologies Extracted from Different Data Sources”

- 2-3 more papers upcoming