Bipartite Internet Topology at the Subnet-level

Mehmet Burak Akgun and Mehmet Hadi Gunes
Department of Computer Science and Engineering, University of Nevada, Reno
Email: {makgun, mgunes}@cse.unr.edu

***Abstract*—Internet topology modeling involves capturing crucial characteristics of the Internet in producing synthetic network graphs. Selection of vital metrics is limited by our understanding of the Internet topology, which relies on the state of the art measurement studies. Recent measurement studies indicate that underlying subnetworks, multi-access links providing one-hop connectivity to multiple nodes, are an important factor shaping the topology of the Internet. Current network models utilize point-to-point edges that can connect exactly two vertices of the topology. Decomposition of the underlying multi-access links into pairwise point-to-point edges results in cliques and is an oversimplification of the analyzed networks. Accurate modeling of multi-access links require special type of edges, i.e. hyper-edges, that can connect multiple vertices in a hyper-graph. Hyper-graphs are best illustrated as bipartite graphs where hyper-edges connect two types of vertices, i.e., router interfaces and subnets. In this paper, we introduce a bipartite model of the Internet topology and discuss representative synthetic network generation.**

Keywords: 2-mode, Internet topology, Hyper-graph

I. Introduction

Modeling real-life networks require a thorough understanding of fine grained and large scale connectivity characteristics of the underlying topology. Missing certain features may result in loss of representations and studies based on these models may falsify our understanding of the real topology. Network modeling is challenging because of the correct identification of the underlying connectivity/growth mechanisms in the construction of the network and of the crucial large scale topological metrics. As the primary goal of network models is to mimic the actual topology in regard of characteristics under analysis, complexity of models directly rely on the level of targeted representativeness. Many complex networks can be modeled as a set of nodes and point-to-point edges as this level of representation is sufficient for practical use cases. However, for certain networks such simple graph modeling approach may miss crucial characteristics.

Router level Internet topology measurement studies deal with sampling the actual topology using various active and passive measurement techniques where the main goal is to reveal the connectivity among routers. Routers typically have more than two interfaces where each interface attaches to a different subnet to route incoming packets. One of the primary challenge stems from the the fact that a single router may be observed with different IP addresses via different measurement probes as each network interface of a router has a unique IP address. This requires the discovery of the set of IP addresses that belong to the same router, namely *IP alias resolution* process, and researchers have proposed multiple approaches to detect the IP aliases [1], [2]. Another issue is the discovery of the actual links that connect the routers. Routers are connected over a subnet that provides one hop connectivity to all interfaces attached to the subnet. While point-to-point links connect exactly two interfaces multi-access links connect multiple interfaces over the same collision domain. Hence, in modeling Internet topology, we need to identify underlying physical subnets among the IP addresses [3], [4].

In a recent study, we point that the interface and subnet distributions follow power-law in addition to observed degree distribution [5]. In this paper, we propose to replicate fine grained features of the Internet via interface and subnet distributions while capturing large scale characteristic of the Internet via observed degree distribution.

In the rest of the paper, we first present our Internet topology model in Section II. In Section III, we present how we match power-law distributions to produce intended topologies with a specified size. In Section III, we present bipartite synthetic topology generation approach. In Section V, we analyze the dependence between fine grained metrics and large scale metrics. In Section VI, we analyze the feasible range of parameter space for obtaining a connected graph and present a minimum spanning tree based network generator that ensures graph connectivity between all nodes. Finally, we conclude the paper in Section VII.

II. Internet Topology Model

Initially, topology generation processes used to replicate the underlying network deployment mechanisms (such as GT-ITM [6] and IGEN [7]) but shown to miss large scale characteristics as they focused solely on the fine grained features. With power-law characterization of Internet degree distribution, generators (such as BRITE [8] and Inet [9]) focused on large scale features but they were not practical [10]. Moreover, focusing on dk-series graphs it is shown that one can match any set of features of observed graphs (such as Orbis generator [11]). However, as the number of features increase matching process becomes infeasible.

In this paper, we present a middle ground that captures both *fine grained* and *large scale* features by focusing on the 2-mode property of the Internet topology[1]. As Internet topologies consist of layer-2 (i.e., switches) and layer-3 (i.e., routers) devices, we can better model the Internet topology by considering the underlying connectivity medium, i.e., multi-access links deployed by switches. This topology should be modeled as a 2-mode graph instead of a conventional 1-mode graph.

2-mode graphs are typically represented with a hyper-graph. A hyper-graph *H=(N,S)* is a generalized graph form where N is the set of nodes and S is the set hyper-edges. Each element of S represents subsets of N which are connected through the same link. Hyper graphs are often illustrated using bipartite graphs.

[1]In our extensive investigation, we have not came across a generator that analyzed 2-mode feature of the link/router-level Internet topology.

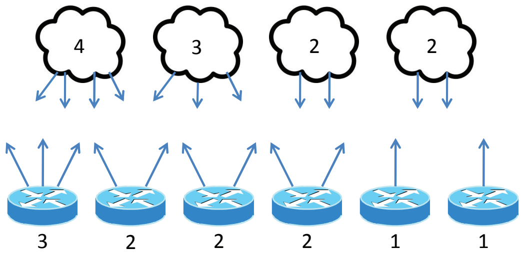

Fig. 1. Sample Bipartite Graph of Interfaces and Subnets

In order to maintain the subnet relations among the routers, we use an undirected bipartite graph where vertices are either interfaces or subnets, as shown in Figure 1. Each interface is attached to at least one subnet and each subnet is attached to at least two interfaces. The number of attachments for each subnet and interface is shown on the figure.

III. Matching Power-law Distributions

In order to generate synthetic topologies that reflect the major characteristics of the Internet at both fine grained and large scale, we analyzed few crucial distributions. At *large scale*, most topology generators capture the degree distribution of the Internet [8], [9]. At *local scale*, we identified node interface distribution and subnet size distribution to be crucial characteristics to capture [5].

In our earlier study [5], we analyzed data from major large scale Internet topology measurement projects and identified power-law distribution in all three distributions. Hence, in this paper, we present a new approach to match power-law distributions in parallel.

Power-law distributions follow the exponential form $\mathbf{F_i} = \mathbf{A}\mathbf{i}^{-\alpha}$ where α is the slope of the graph and $\mathbf{A}$ is the coefficient which shifts the graph in log scale. Power-law graphs are usually illustrated in logarithmic scale. As indicated in Equation 1, we observe that in log-log scale, power-law distributions are linear and exponent α determines the slope of the linear curve.

$$\log(F_i) = \log(Ai^{-\alpha}) = \log(A) + (-\alpha)log(i) \quad (1)$$

In generating synthetic topologies, we modify the scaling coefficient $\mathbf{A}$ to obtain different size power-law distributions with the same slope α. As integral of the distribution yields the network size, we can adjust $\mathbf{A}$ to obtain a power-law distribution with a targeted size [12]. Figure 2 illustrates the scaling of $\mathbf{A}$ which simply shifts the curve upwards or downwards depending on the desired network size. Note that, as the curve itself is not linear in regular scale, the maximum degree in the network does not scale linearly [11].

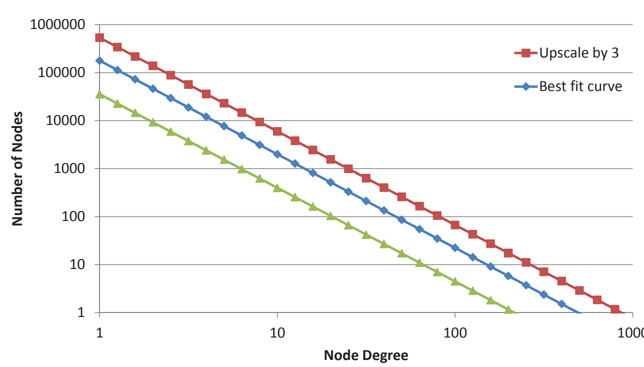

Fig. 2. Power-law curve Shift

IV. Synthetic Topology Generation

While generating synthetic topologies, we rely on 3 distributions that can be specified by the power-law exponents α. Slope of interface distribution $\alpha_{\mathbf{ID}}$, subnet distribution $\alpha_{\mathbf{SD}}$, and observed degree distribution $\alpha_{\mathbf{DD}}$ can be either provided by the user or obtained from measurement data. Once the user specifies a target network size, i.e., number of node $\mathbf{NN}$, we can determine the distributions by calculating suitable $\mathbf{A}$ coefficients and generate a synthetic topology.

Given the slope $\alpha_{\mathbf{ID}}$, the area beneath the interface distribution curve should match the number of nodes $\mathbf{NN}$.

$$NN = \sum_i ID_i \quad (2)$$

For instance, in Figure 1, there are 1 node with 3 interfaces ($ID_3 = 1$), 3 nodes with 2 interfaces ($ID_2 = 3$), and 2 nodes with 1 interface ($ID_1 = 2$). Hence, we can compute the scaling coefficient $\mathbf{A_{ID}}$ of the interface distribution $\mathbf{ID_i}$. Once the interface distribution is determined, the number of interfaces can be calculated from Equation 3.

$$NI = \sum_i iID_i \quad (3)$$

As the number of interfaces on all nodes is equal to the sum of subnet sizes (see Equation 4) and the slope of the subnet distribution $\alpha_{\mathbf{SD}}$ is given, we can determine the scaling coefficient $\mathbf{A_{SD}}$ of the subnet distribution.

$$NI = \sum_j jSD_j \quad (4)$$

The total number of subnets is calculated with Equation 5. Finally, the degree distribution is computed with Equation 6.

$$NS = \sum_j SD_j \quad (5)$$

$$NN = \sum_k DD_k \quad (6)$$

Exact scaling of the distributions can be quite difficult to implement due to the discrete nature of the generated graphs and rounding of integers. After calculating all distributions, we generate vertices of the bipartite graph without any edges.

TABLE I. Notation

NN	Number of Nodes
NS	Number of Subnets
NI	Number of Interfaces
α_{ID}	Slope of Interface Distribution
α_{SD}	Slope of Subnet Distribution
α_{DD}	Slope of Degree Distribution
ID_i	Interface Distribution, i $\in$ [1, Max interface size]
SD_j	Subnet Distribution, j $\in$ [2, Max subnet size]
DD_k	Degree Distribution k $\in$ [1, Max degree]
$Node_n$	n^{th} node where n $\in$ [1,NN]
$Subnet_s$	s^{th} subnet where s $\in$ [1,NS]
IC_n	Desired Interface Count of Node-n
SS_s	Desired Subnet Size for the Subnet-s

We assign the size of each subnet $\mathbf{SS_s}$ and the interface count of each node $\mathbf{IC_n}$ according to the previously calculated $\mathbf{SD_j}$ and $\mathbf{ID_i}$, respectively.

We then mix the order of the nodes and subnets to eliminate any bias. We may alter the assortative mixing [13] by ordering subnets and nodes are with respect to their size or another feature. However, measured data sets were non-assortative in terms of attachment between low/high interfaces and subnets [5].

In the second phase of the generation, we attach nodes to the subnets as in Algorithm 1. To reduce the complexity, we initially allow multiple attachments between the same node-subnet pair.

Algorithm 1 Edge Attachment

```
s ← 1
for all n such that 1 ≤ n ≤ NN do
    for all x such that 1 ≤ x ≤ IC_n do
        while !Is-Attachable(node-n, subnet-s) do
            s ← (s + 1)%NS
        end while
        Attach(node-n, subnet-s)
    end for
end for
```

In the algorithm, *Is-Attachable* function checks whether $\mathbf{subnet-s}$ has reached its target subnet size $\mathbf{SS_s}$ and returns true if there is room. If true, *Attach* function creates an edge between $\mathbf{subnet_s}$ and $\mathbf{node_n}$. After attaching all nodes to subnets, we check whether a subnet-node pair is linked multiple times (code not shown). In multi-attachment cases, we choose a random node and swap their subnets.

At this phase of the algorithm, network fully satisfies the $\mathbf{ID_i}$ and $\mathbf{SD_j}$ distributions. However, the $\mathbf{DD_k}$ might be a bit off of the desired distribution. Hence, we designed *heterogeneous swap* and *homogeneous swap* functions similar to the rewiring process used in graph generation algorithms [12].

Heterogeneous swap function is used to tune the degree distribution of a graph without affecting the previously met $\mathbf{ID_i}$ and $\mathbf{SD_j}$ distributions. The function intelligently selects two nodes and an interface of each of these nodes such that $\mathbf{DD_k}$ distribution converges. In particular, it chooses a node with a degree that is above the curve and another with a degree below the curve. Then, it picks interfaces that would converge the $\mathbf{DD_k}$ distribution when their subnets are swapped. Note that, even thought the swap process is random, if parameters are feasible (see Section VI), it will eventually converge to the targeted distribution.

Homogeneous swap is similar to heterogenous swap but it only selects interfaces that are connected to the same size subnets. Homogeneous swap has no effect on $\mathbf{ID_i}$, $\mathbf{SD_j}$ or $\mathbf{DD_k}$ distributions. It may be utilized to connect a partitioned network or to tune the average path length of the network.

V. Correlation Between Distributions

Given $\mathbf{NN}$, $\alpha_{\mathbf{ID}}$, $\alpha_{\mathbf{SD}}$, and $\alpha_{\mathbf{DD}}$, described procedure generates synthetic topologies with the desired distribution characteristics. However, our evaluations indicate that $\mathbf{ID}$, $\mathbf{SD}$, and $\mathbf{DD}$ are not completely independent. For example, if $\alpha_{\mathbf{ID}}$ is set lower than $\alpha_{\mathbf{DD}}$, the heterogeneous swap can never converge to the desired degree distribution.

Our analysis reveal the dependency of the degree distribution $\mathbf{DD_k}$ over the interface $\mathbf{ID_i}$ and the subnet $\mathbf{SD_j}$ distributions. Once interface and subnet distributions are determined (for given $\alpha_{\mathbf{ID}}$ and $\alpha_{\mathbf{SD}}$), we can compute the average degree in the network as follows. Since a subnet of size j, by definition, connects $\mathbf{j}$ nodes within one-hop distance, once a node connects to this subnet, its degree increases by $\mathbf{j-1}$. Hence, the total degree contribution of the subnet to the network is $\mathbf{j \times (j-1)}$. When we consider all subnets in the network, the total degree can be computed as;

$$TotalDegree = \sum_{j} SD_j \times j \times (j-1) \tag{7}$$

For a power-law distribution with a certain slope α and pre-determined size $\mathbf{NN}$, average degree uniquely determines the scaling exponent $\mathbf{A}$ as $\mathbf{NI = AverageDegree \times NN}$. Hence, both the scaling coefficient $\mathbf{A}$ and the power law exponent $\alpha_{\mathbf{DD}}$ can uniquely be calculated for given $\mathbf{NN}$, $\alpha_{\mathbf{ID}}$ and $\alpha_{\mathbf{SD}}$ parameters.

VI. Connectivity Analysis

Utilizing pure probabilistic generation methods after assigning certain targeted degrees to nodes is found to perform poorly in terms of ensuring graph connectivity [14]. During our experiments, we observed that random matching of interfaces to subnets results in disconnected graphs and the giant component shrinks as the subnet size distribution and interface distribution becomes more steep. Hence, in this section, we provide a connected version of the graph generation algorithm and analyze the feasible region of parameter space to obtain a connected graph.

As the network is modeled as a bipartite graph, we define a modified minimum spanning tree which guarantees at least a single path between any subnet pairs in Algorithm 2. The design of modified spanning tree algorithm relies on the idea that a subnet of size j can be used to connect up to $j-1$ nodes to the giant component. This can be achieved by attaching the subnet to the current giant component and using the rest of its capacity to connect $j-1$ nodes not in the giant component. Since single interface nodes act like a stub in the network, they do not help with connectivity.

Algorithm 2 Establishing a Minimum Spanning Tree

```
s ← 1
for all n such that 1 ≤ n ≤ NN do
    if IC_n ≥ 2 then
        for all x such that 1 ≤ x ≤ IC_n do
            Attach(node-n,subnet-s)
            s ← (s + 1)
            if s=NS then
                return
            end if
            s ← (s − 1)
        end for
    end if
end for
```

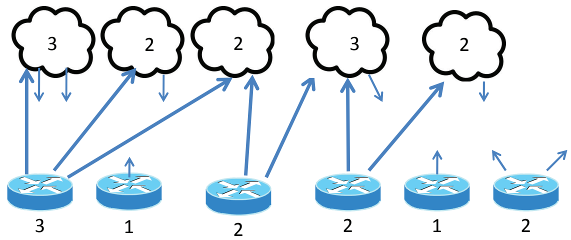

Fig. 3. Subnet Spanning Tree

The algorithm traces every node which has more than one interface. For every interface of these nodes, the algorithm attaches it to a new subnet and iterates the subnet pointer to the next. At each iteration of nodes, we decrement subnet pointer once so that the first attachment of the new subnet is made to the current giant component. When all subnets are attached the algorithm terminates. As subnets and nodes are randomly mixed, this attachment approach does not cause any bias that may change assortavitity.

Figure 3 illustrates the subnet spanning tree which generated at least a single path between any nodes for the sample in Figure 1. Note that, single interface nodes are skipped and algorithm quits whenever every subnet is connected.

As the number of interfaces are fixed for each node, algorithm runs in $O(N)$ where N is the number of nodes. Note that, algorithm does not match every node to a subnet as it builds a spanning tree of subnets. Remaining nodes are attached in the next phase to tune the degree distribution.

Generating a subnet spanning tree is not possible for all $\alpha_{\mathbf{ID}}$ and $\alpha_{\mathbf{SD}}$ values. Intuitively, as the ratio of single interface routers increases, it becomes harder or even impossible to generate a connected network.

We derive a sufficient condition that guarantees the existence of a connected configuration in Equation 8. Every node which has more than one interface can utilize its first interface to attach to the current giant component and each of the rest of its interfaces to attach to a new subnet. Hence, we can define the connectivity parameter based on the Equation 8.

$$NS \leq 1 + \sum_{i=2} (i-1) \times ID_i \tag{8}$$

Figure 4 plots the connectivity parameter with respect to $\alpha_{\mathbf{ID}}$ and $\alpha_{\mathbf{SD}}$ values. The region where connectivity parameter is below 1 (i.e., blue area) is the feasible region for a connected graph. As slopes become more steep, ratio of single interface nodes and ratio of smaller subnets increase and eventually graph connectivity becomes infeasible.

VII. Conclusion and Future Work

In this paper, we present a 2-mode synthetic topology generator that can capture both fine grained and large scale characteristics of the Internet topology. In particular, we match interface and subnet distribution of routers at link-level to accurately represent network connectivity practices while capturing large scale observations in observed degree distribution. For this, we derive formulas that are used in synthetic network generation and provide algorithms that would produce synthetic topologies with intended characteristics.

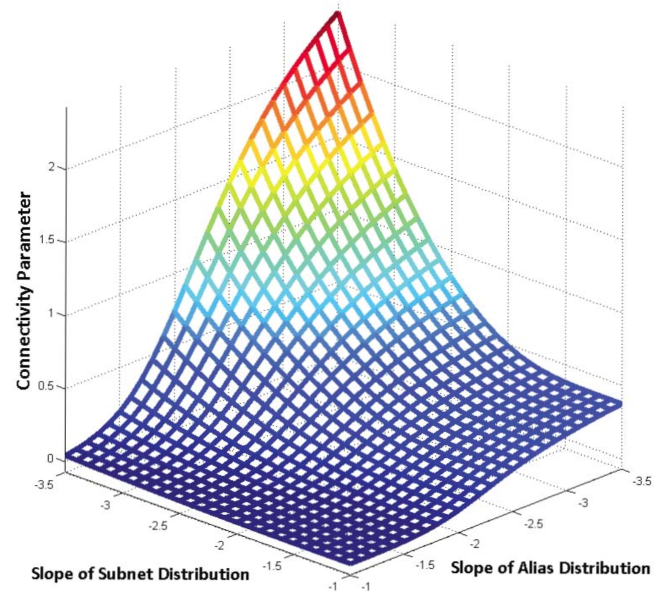

Fig. 4. Connectivity Feasibility

As a future work, we will investigate additional metrics such as characteristic path length, hop distribution, assortativity, and clustering of generated 2-mode graphs. We will also investigate the cost of building generated topologies to analyze their feasibility in real life. Similarly, we will investigate the traffic flow related features of generated networks in addition to topological characteristics.

References

[1] M. Gunes and K. Sarac, "Resolving IP aliases in building traceroute-based Internet maps," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1738 –1751, Dec 2009.

[2] A. Bender, R. Sherwood, and N. Spring, "Fixing ally's growing pains with velocity modeling," in *ACM SIGCOMM IMC* , Vouliagmeni, Greece, Oct 20-22, 2008.

[3] M. Gunes and K. Sarac, "Inferring subnets in router-level topology collection studies," in *ACM SIGCOMM IMC*, San Diego, CA, Oct 24-26, 2007.

[4] M. Tozal and K. Sarac, "Subnet level network topology mapping," in *IEEE IPCCC*, Orlando, FL, Nov 17-19, 2011.

[5] M. Akgun and M. Gunes, "Impact of multi-access links on Internet topology modeling," *ACM SIGCOMM CCR*, (under review).

[6] M. Thomas and E. W. Zegura, "Generation and analysis of random graphs to model internetworks," Georgia Tech, Tech. Rep., 1994.

[7] B. Quoitin, "Towards more representative internet topologies," Universit Catholique de Louvain, Tech. Rep., 2010.

[8] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: Universal topology generation from a user's perspective," Boston University, Tech. Rep., 2001.

[9] J. Winick and S. Jamin, "Inet-3.0: Internet topology generator," University of Michigan, Tech. Rep., 2002.

[10] D. Alderson, L. Li, W. Willinger, and J. Doyle, "Understanding Internet topology: principles, models, and validation," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1205 – 1218, Dec 2005.

[11] P. Mahadevan, C. Hubble, D. Krioukov, B. Huffaker, and A. Vahdat, "Orbis: rescaling degree correlations to generate annotated internet topologies," *ACM SIGCOMM CCR*, vol. 37, pp. 325–336, Aug 2007.

[12] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.

[13] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, no. 20, pp. 208 701+, Oct 2002.

[14] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat, "Systematic topology analysis and generation using degree correlations," *ACM SIGCOMM CCR*, vol. 36, pp. 135–146, Aug 2006.