An Comparative Evaluation of Graph Metrics in Measuring the Resilience of Telecommunications Backbone Networks

CS224W: Analysis of Networks - Project Final Report
Stanford University

Ramesh Subbaraman
rameshrs@stanford.edu

1 Introduction

Today, the telecomminications backbone networks that underlie the internet and cellular networks have become indespensible to the life-styles of a large fraction of the world's population and a significant fraction of commercial activities happen over such networks. This is only expected to increase in the future. Thus, the resilence of these networks to disruptions caused by failures as well as attacks that target them is an important issue. While several metrics have been proposed in the graph theory and network science literature for analyzing the resilience of networks in general, as well as by researchers in application domains where problems can be modeled as networks; it is not entirely clear which of them if any can be best used to model network resilience in the specific application domain of telecommunication backbone networks since ultimately the definition of what resilience means can be very application specific. In the telecommunications backbone context, network providers invest in purchasing capacity to carry traffic; and the revenue generated is proportional to the amount of traffic carried. Disruptions to traffic flows can occur due to accidents for e.g. caused by fibre cuts, power outages, mis-configurations etc.; or deliberate attacks e.g. denial of service attacks, acts of terror, disgruntled employee etc. Thus, people planning, designing and running these networks are interested in having measures of their reliability in terms of carrying revenue generating traffic. In this project, we propose to further examine the resilience of such networks by evaluating a set of 14 graph metrics in terms of their correlation to the resilience of telecomminucations backbone networks, where we define our measure of resilience in terms of the demand (i.e. traffic) carrying capability of the network. We conduct evaluations against several types of random failures and targeted attacks, and use not only synthetic but also real-world networks for our experimental analysis.

2 Related Work

Reference [1] focuses on the resilience service oriented networks, where servers are capable of offering some set of services locally by themselves and can serve as relay nodes for remote services offered by other nodes. The authors point out that the traditional metrics measuring network resilience in terms of graph partitioning are ill-suited for their application since from a service availability point of view if each sub-graph that results from the partition has nodes offering all services for which there is demand within the sub-graph , the network is still fully functional ("self-sufficient") for their purposes. Thus, they define new resilience metrics edge(node)-failure-resilience which are the largest integer k such that no matter which subset of k or fewer edges (nodes) fail, each sub-graph that results remains self-sufficient; and then develop polynomial time algorithms to measure the above 2 metrics. However, they do not perform any comparative analysis with other metrics or experimental evaluation of the proposed resileince metrics' efficacy. References [2] and [3] investigated the resilience of supply networks, wherein there are multiple types of nodes - supply nodes, demand nodes, and hybrids; and this application too has a notion of demand that must be satisfied which could be possible even if failures cause the network to be partitioned (e.g. a supply node is in every sub-network then they are all "functional" sub-networks). They present metrics and evaluate them on synthetic networks against random failures as well as targeted attacks wherein high degree nodes were removed. However, from the point of view of network resilience of a telecommunications backbone network, the models, metrics and measures of resilience used in these studies from other application areas are ill-suited. Specifically, there is no notion of the size or value of the demand, so the metrics and measures of success can for example claim low resilience even if miniscule amounts of demands get disrupted. Also, the edges in their network are un-weighted whereas in telecommunications backbone networks edges are capacitated, and the metrics and measures of resilience presented only care about the existence of a path under failures and not whether a path of adequete capacity is available under failures.

In the telecommunications context, [4] looked at an Autonomous System (AS) level graph of the Internet and analyzed its resilience again in terms of connectivity (size of largest component) and path lengths (diameter). However, unless modeling the effects of ISPs shutting down or having misconfigurations of the Border Gateway Protocol on their routers, it is not clear that the AS level network is appropriate to measure telecommunications network resilience in terms of even connectivity let alone demand carrying capability. This is because it seems like the network data was an undirected simple graph with a single link between ASes that connect to each other, whereas ASes that are large ISPs for example will peer with each other at several locations thus increasing the resilience of the network relative to what the authors' model assumes. Furthermore, the existence of a link does not imply that all kinds of traffic can actually flow on it since usually enterprise ASes that are customers of multiple ISPs (possibly at different geographic locations) will not and / or cannot route traffic between the ISPs in the event of a failure, which reduces the resilience relative to what the authors' analysis based on random failures and targeted removal of high degree nodes would suggest. On the other hand, [5] use a router level graph, propose new metrics (effective diameter, hop exponent, effective eccentricity) and evaluate them as measures of resilience in terms of connectivity (number of nodes that can reach each other). Here again, there is no notion of capacity and the measure of resilience is not demand carrying capability. Also, the router level granularity is likely to make link failures look like they have no impact since most routers in the network will likely be located geographically close to their neighbours in a local area network, and have several (cheap) redundant links within their cluster - which is not relective of what we could expect if we look at wide area networks consisting of (expensive) long haul links which constitue the backbone of the internet. Also, the process for collecting a router level graph of the internet suffers from data accuracy issues - for example the maximum degree in the network is reported to be in excess of 2000 and it is inconceivable even today let alone more than 15 years ago when the paper was published, and this is probably because some networks have implemented security policies to prevent mapping of their networks and a single security device replies to all probe messages.

The study most closely related to ours is [6] which compared the correlation of 18 graph metrics to the resilience of computer networks under targeted attacks. Towards that end, it proposed *sum of flows robustness* which measures the sum of the ratio of the number of nodes that can reach one another to the total number of node pairs in the network as the measure of resilience, and using 3 kinds of targeted attacks based on successively removing nodes with (a) highest degree, (b) highest betweenness (i.e. number of shortest paths going via the node) and (c) highest closeness (i.e. lowest mean hop count to other nodes) compares the metrics' correlation to their network resilience measure. However, the paper left modeling weighted edges (i.e. capacity) and running evaluations against real network topologies as future work; and it is not clear that the measure of success chosen i.e. sum of flow robustness, is the appropriate one for communications networks since for example admitting connectivity between a large number of small demand nodes as opposed to a few nodes with large amount of demand may not be seen as a success in this application domain. Also, only targeted attacks with important nodes being removed was explored - in a backbone network it is more likely that failures of or successful attacks against edges (e.g. fibre cuts) rather than entire nodes (routing complex / data center) occur, so it would have been meaningful to also have some link failure and attack scenarios.

Unlike these approaches, in our work we model link capacities, define a measure of network resilience based on demand carrying capability of the network, and include real internet backbone networks. Furthermore, we also include some previously unevaluated graph metrics, and consider link failures and attacks in our experiments.

3 Our Model

In this section we describe our network model for a demand carrying capacitated telecommunications backbone network, state/define the 14 graph metrics whose correlation to our measure of network resilience we will compare along with any graph algorithms needed to compute them, and our measure of network resilience in terms of the network's demand carrying capacity.

3.1 Network Model

We model a telecommunications backbone network as an undirected graph where the node set V represents a computing complex (this could be co-located with a data center) and edge set E represents communication links. The edges are weighted with the weights representing link capacities, i.e. for each edge $(u, v) \in E$ we have a non-negative capacity, c(u,v). Also, between each pair of nodes in the network $u, v \in V$ we have a demand D(u,v) that is to be carried over the network from u to v, and is measured in the same units as the link capacities.

3.2 Graph Metrics and Algorithms

Based on the results presented in [6], we choose the following 6 of their graph metrics for our comparative evaluations:

1. Algebraic Connectivity, AC: the second smallest eigenvalue of the Laplacian matrix, L, of the graph; where the Laplacian matrix is defined as [7]

$$L_{u,v} = \begin{cases} deg(u), & \text{if } u = v \\ -1, & \text{if } (u,v) \in E \\ 0, & \text{otherwise} \end{cases}$$

2. Network Criticality, NC = $\frac{2}{|V|-1} * Trace(L^{-1})$; where L^{-1} stand for the inverse of L.

3. Effective Graph Conductance, EGC = $\frac{|V|-1}{N * \sum_{i=2}^{|V|} \frac{1}{\lambda_i}}$; where λ_i are the eigenvalues of the L sorted in non-decreasing order.

4. Natural Connectivity, NaC = $\ln(\frac{1}{|V|} * \sum_{i=1}^{|V|} e^{\delta_i})$; where δ_i are the eigenvalues of the adjacency matrix of the graph.

5. Variance of Node Betweenness, σ_{NB} i.e. the variance of the vector of node betweenness values, where betweenness of a node is the number of shortest paths (measured as per hop-count i.e. all edge weights 1) in the graph passing through the node.

6. Variance of Link Betweenness, σ_{EB} i.e. the variance of the vector of edge betweenness values, where betweenness of an edge is the number of shortest paths (measured as per hop-count i.e. all edge weights 1) in the graph using the edge.

We further consider the following 8 metrics which capture in some sense the path diversity and/or capacity in the network:

7. Mean of Node Betweenness, μ_{NB} i.e. the mean of the vector of node betweenness values (again measured as per hop-count).

8. Mean of Link Betweenness, μ_{EB} i.e. the mean of the vector of edge betweenness values (again measured as per hop-count).

9. Mean Number of Node Disjoint Paths, μ_{NDP} i.e. the mean of the number of node disjoint paths in the graph between every pair of distinct nodes in the graph.

10. Variance of Number of Node Disjoint Paths, σ_{NDP} i.e. the variance of the number of node disjoint paths in the graph between every pair of distinct nodes in the graph.

11. Mean Number of Edge Disjoint Paths, μ_{EDP} i.e. the mean of the number of edge disjoint paths in the graph between every pair of distinct nodes in the graph.

12. Variance of Number of Edge Disjoint Paths, σ_{EDP} i.e. the variance of the number of edge disjoint paths in the graph between every pair of distinct nodes in the graph.

13. λ_{min}^{CTNC}: the smallest non-0 eigenvalue of the capacity matrix (i.e. weighted adjacency matrix) except with diagonal entries containing the total outgoing capacity from a node.

14. λ_{min}^{NDTND}: the smallest non-0 eigenvalue of the matrix containing the number of node disjoint paths between nodes, except with diagonal entries containing the total outgoing number of node disjoint paths from a node to other nodes.

The metrics 1 - 4, and 13 and 14 can be easily computed using linear algebra. For the node and edge betweenness related metrics (5 - 8), we used the functions provided by the Stanford Network Analysis Project, SNAP-PY [8] package with the appropriate parameter set to ensure that an exact value rather than an approximation is returned. Metrics 9-12 require computing the number of edge and node disjoint paths between all pairs of distinct nodes in the graph, which can be done with a well known algorithm described next.

For a given pair of distinct nodes s and t in G, the number of edge disjoint paths between them can be computed by solving a max-flow problem for s to t on a graph that is the same as the given graph except with all edge capacities set to 1 [9] in which case the max-flow value is the number of edge disjoint paths. Specifically, we define a flow[1] f(u,v) $\forall u, v \in V$ as a non-negative number such that $f(u, v) \leq c(u, v)$, and $\forall v \in V, \sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$; then the max-flow problem for s to t can be solved using the Ford-Fulkerson algorithm as shown in Figure 1 [10], where residual capacity is computed as c'(u,v) = c(u,v) - f(u,v) + f(v,u), and residual network is a graph with link capacities equal to the residual capacities. If we solve this problem repeatedly for every distinct node pair, we get the number of edge disjoint paths for all distinct pairs of nodes.

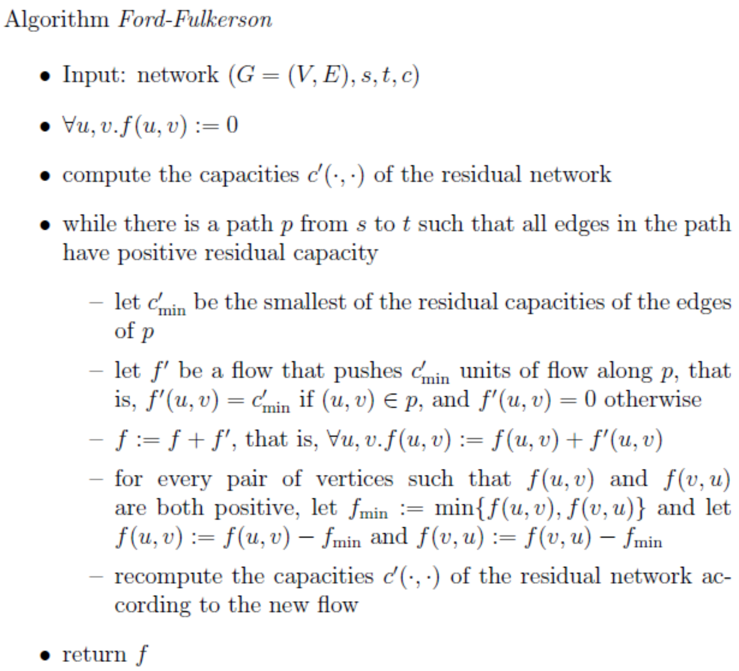
For node disjoint paths between 2 distinct nodes s and t in G, we can reduce the problem to one of edge-disjoint paths by doing as follows [11] - create a new directed graph where every node u in the original graph has been split into 2 nodes, u_{in} and u_{out}, with an edge from u_{in} to u_{out}; and every edge (u,v) in the original graph is replaced by edges from u_{out} to v_{in}. Then finding the number of edge-disjoint paths from s_{out} to t_{in} in the new graph using the algorithm in Figure 1, gives us the number of node disjoint paths between s and t in G.

We will compare the 14 metrics in terms of their correlation to our network resilience measure (described in the next subsection) by using the Spearman rank correlation coefficient which is a value $\rho \in [-1, 1]$ with the endpoints indicating perfect negative and positive correlations respectively [6]. Thus, the metrics which have the highest absolute value of ρ for a particular failure or attack across all the graphs we consider, "win" for that failure or attack.

3.3 Measure of Resilience

We define the measure of network resilience for our use case as the sum across the sequence of network states considered of the ratio of the total demand that was routed (i.e delivered) to the total demand. Specifically, fraction of the total demand routed under a particular network state k, $FR_k = \frac{\sum_{u,v \in V} DR_k(u,v)}{\sum_{u,v \in V} D(u,v)}$, where $DR_k(u, v)$ is the demand from u to v that got

[1] This is the conventional definition of a flow, however [6] uses the term differently inthe context of "flow robustness" defined earlier. Except when referring to flow robustness, in this work the term flow is used as defined here.



Algorithm *Ford-Fulkerson*

- Input: network $(G = (V, E), s, t, c)$
- $\forall u, v.f(u, v) := 0$
- compute the capacities $c'(\cdot, \cdot)$ of the residual network
- while there is a path p from s to t such that all edges in the path have positive residual capacity
  - let $c'_{\min}$ be the smallest of the residual capacities of the edges of p
  - let f' be a flow that pushes $c'_{\min}$ units of flow along p, that is, $f'(u, v) = c'_{\min}$ if $(u, v) \in p$, and $f'(u, v) = 0$ otherwise
  - $f := f + f'$, that is, $\forall u, v.f(u, v) := f(u, v) + f'(u, v)$
  - for every pair of vertices such that $f(u, v)$ and $f(v, u)$ are both positive, let $f_{\min} := \min\{f(u, v), f(v, u)\}$ and let $f(u, v) := f(u, v) - f_{\min}$ and $f(v, u) := f(v, u) - f_{\min}$
  - recompute the capacities $c'(\cdot, \cdot)$ of the residual network according to the new flow
- return f

Figure 1: Solving a Max-flow Problem

routed under network state k; and our measure of resilience for a specific run of a simulation (see the Experiments section) consisting of K iterations is $\sum_{k=0}^{K} FR_k$, where k = 0 refers to the initial state.

To determine the value of $\sum_{u,v \in V} DR_k(u, v)$, we will create a graph G_k by setting capacities $c_k(u, v)$ of all edges (u,v) absent in the current network state k set to 0; then defining $f^{s,t}(u, v)$ as a flow from source s to destination t being carried on edge (u,v), solve a fractional multi-commodity flow problem modeled as a linear program shown in Figure 2 using the Python Mathematical Programming Package [12] with these updated capacities. The linear program maximizes the total flow leaving sources subject to the following constraints (1) link capacity cannot be exceeded, (2) all flows entering a non-source or non-destination node must leave the node, (3) all flow leaving a source must arrive at the destination, (4) the total flow leaving the source cannot be larger than the demand, (5) a flow cannot enter its source, (6) a flow cannot leave its destination; and (7) all flows must be non-negative. Then, the optimal objective value obtained is the desired value, i.e. the total demand routed in network state k. Post processing the outputs of the solver will give us the paths taken by each flow, which will be needed for executing some of the experiments (see the Experiments section).

$$\max \sum_{(s,t)\in E} \sum_{(s,v)\in E} f^{s,t}(s, v)$$

$$s.t. \sum_{(s,t)\in E} f^{s,t}(u, v) \leq c_k(u, v) \qquad \forall (u, v) \in E \qquad (1)$$

$$\sum_{(u,v)\in E} f^{s,t}(u, v) = \sum_{(v,w)\in E} f^{s,t}(v, w) \qquad \forall (s, t) \in E; \quad \forall v \in V - \{s, t\} \qquad (2)$$

$$\sum_{(s,v)\in E} f^{s,t}(s, v) = \sum_{(u,t)\in E} f^{s,t}(u, t) \qquad \forall (s, t) \in E \qquad (3)$$

$$\sum_{(s,v)\in E} f^{s,t}(s, v) \leq D(s, t) \qquad \forall (s, t) \in E \qquad (4)$$

$$f^{s,t}(u, s) = 0 \qquad \forall (s, t) \in E; \quad \forall (u, s) \in E \qquad (5)$$

$$f^{s,t}(t, v) = 0 \qquad \forall (s, t) \in E; \quad \forall (t, v) \in E \qquad (6)$$

$$f^{s,t}(u, v) \geq 0 \qquad \forall (s, t) \in E; \quad \forall (u, v) \in E \qquad (7)$$

Figure 2: Linear Programming Formulation of the Fractional Multi-Commodity Flow Problem

4 Data Sets

We conduct experiments on 2 real world networks of large internet service provides- a US backbone network Cox [13], and NTT's international backbone network [14]. By manually transcribing the images of the network maps for the 2 providers,we created text files containing space separated (source node, destination node, capacity) tuples [2] (multi-edges were collapsed into a single edge with capacity set equal to the sum of each edge). The number of nodes and edges for the Cox and NTT networks were 32 and 60, and 40 and 81 respectively. Since the data was manually transcribed which is error prone, for both networks we did some due diligence checks - before replacing node names in our original transcription with node numbers for use in our experimentations we printed out the set of node names (cities) to verify that typographical errors had not added any non-existent nodes, we plotted the edge capacities to make sure that the numbers were within an acceptable range, and used SNAPPY[8] to check that the graphs were connected.

In the interest of comparing to null models, we also generated an Erdos Reyni (ER) random graph and a Barabasi-Albert (BA) scale free graph based on the number of nodes and edges in the Cox network. Specifically, we used SNAPPY[8] functions, and the ER graph generator was given the Cox number of nodes and edges as input parameters, and the BA graph generator was given the number of nodes in Cox and $\left\lfloor \frac{\text{number of edges in Cox}}{\text{number of nodes in Cox}} \right\rfloor$ as input parameters. To help provide intuitions and help reasoning about results obtained from our metrics and algorithms, we also use 4 highly structured graphs - circle, wheel, star and ladder [6] with 40 nodes like the NTT network. For these synthetic networks (generated and structured), we set link capacities drawing uniformly at random from the set {10, 20, 40, 100, 120, 140, 200} which is similar to the real networks except we did not use very large capacities such as 400 in the interest of the run-time of the linear programming solver which has to be called repeatedly in our experiments (since our measure of success is a ratio of demand routed to tota demand, this scaling does not affect our ability to compare results across networks).

For each network we generated a demand set as follows. We initially set the demand between every pair of nodes to be a random integer in {a, b}, with a = 10, b = 50 for the real networks, and a = 2, b = 12 for the synthetic networks. We then solved the fractional multi-commodity flow problem modeled as a linear program, to determine what part of each demand got routed (overall the fraction routed was never more than 0.5). We then created the final demand by re-setting demand values in the original set to what was actually routed, except in cases when the demand routed was 0 and the associated node pair had a direct link between them (in which case we left the demand as is). The reason we did not re-set demands to zero between directly connected node-pairs (hence the final demand fraction was not 1) is these situations is that in if we did in network states when those 2 nodes end up being in a component, despite there being capacity avialable we would not have any demand between them to use that capacity which can bias our results. We then re-ran the linear program and the resulted showed that in the final demand set for every network the fraction of demand routed was larger than 0.968. Since our evaluation is really about how the fraction of demand routed decreases under failures and attacks, a small amount of demand not being routed at the beginning when all links are up is acceptable.

5 Experiments

5.1 Random Failure and Targeted Attack Models

In a single experiment scenario, we run a simulation of a particular type of random failure or targeted attack against a particular network. The failure or attack will consist of iteratively eliminating a particular entity (node or link) from the network chosen according to a criterion which in the targeted attack cases is a measure of the importance of the entity within the network. Specifically, we ran simulations of the following random failure scenarios (1) random node elimination (node_rand), and (2) random link elimination (link_rand); where the probability of being removed is uniform for all entities. For the targeted attack scenarios we ran simulations of the following elimination citerion from [6]: in the next iteration of a simulation run we (1) eliminate the node that currently has the highest degree (node_deg), (2) eliminate the node that currently has the highest node betweenness with shortness measured as per hop-count (node_bc), and (3) eliminate the node that currently has the highest closeness i.e. lowest mean hop count to other nodes (node_close), all of which can be done using SNAPPY [8] functions. We further also simulate the following targeted attack criteria: in the next iteration eliminate (4) the currently most utilized [3] link (link_util), (5) the link that currently carries the largest count of flows (link_count), and (6) the link that currently has the highest link betweenness (link_bc), and (7) the link with the highest capacity (link_cap).

For any given random failure simulation run, our stopping criterion will be 10 iterations; and we always run 5 such simulations and average results on a per iteration basis. For targeted attack simulation runs, our stopping criterion is either the reduction of the fraction of the total demand routed, FR_k to being less than or equal to 20%, or the fraction of remaining entities (corresponding to the attack type) being less than or equal to 10%; or 50 iterations, whichever comes first.

[2] These have been uploaded to http://snap.stanford.edu/submit in a single zip file as the answer to question 3 under Final Report.

[3] utilization(u,v) = $\frac{\sum_{s,t \in V} f^{s,t}(u,v)}{c(u,v)}$

5.2 Results

In this subsection, based on our experimental results we first demonstrate the usefulness of our new measure of resilience based on demand carrying capacity relative to node-pair connectivity based measures that have been traditionally used. We then present our comparative analysis of the 14 graph metrics' correlations to random failures and targeted attacks.

5.2.1 Comparison of Measures of Resilience

Among the measures used by other researchers to evaluate internet networks (see Related Work section), we choose flow robustness [4][6] as the representative metric for our comparisons since it encompasses most of the other measures reported (e.g. number of reachable pairs [5] is the numerator, while relative size of the largest cluster [4] is the first term in the summation) and like our metric is also on a [0 1] scale. Figure 3 shows the evolution across iterations of our fraction of demand routed resilience metric (the first row of figures) for structured graphs against (going left to right) random failures, node targeting attacks, and link targeting attacks [5]. The corresponding flow robustness metric evolution is plotted in the bottom row. The leftmost legend applies to the random failures data (both top and bottom plots), the middle to node targeting attacks and the righmost to the link targeting attacks. Note that since the attacks are split by entity being eliminated, the entity name has been dropped from attack names in the legend. Figure 4 has the same structure except it shows the results for the 2 real networks and the ER and BA graphs instead of the structured graphs.

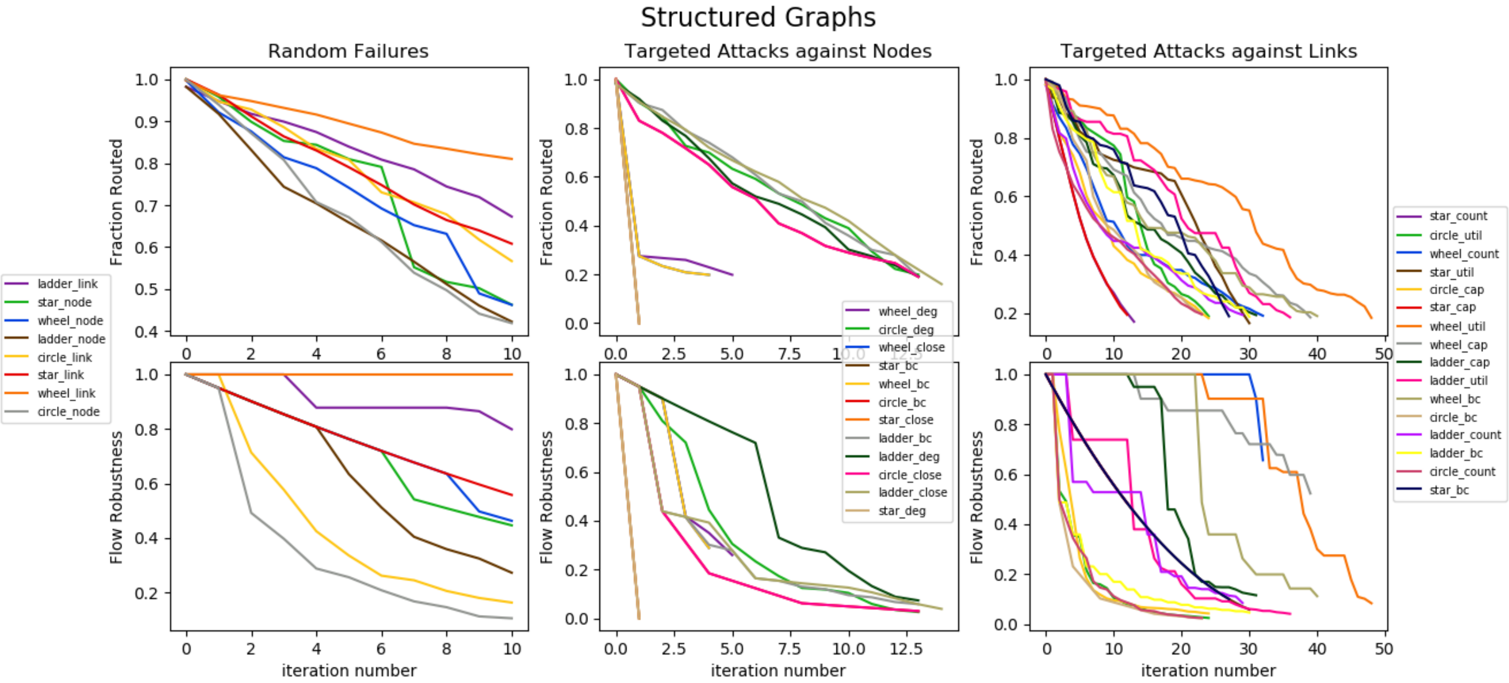


Figure 3: Results for Structured Graphs

As seen for example in Figure 3, the trends of the 2 measures of resilience can be very different - e.g. the wheel graph under random link failures the curve for flow robustness stays put at 1.0 across 10 iterations since eliminating random links does not easily disconnect a node pair in a wheel graph (there are 3 edge disjoint paths between every pair); however the loss of capacity from the network does take away about 20% of the network's demand carrying capacity as seen from the fraction routed curve. The circle graph under random node failures on the other hand shows the opposite trend, whereby the flow robustness drops rather quickly (e.g. about 0.5 after 2 iterations, about 0.3 after 4) reflecting the fact that every 2 node failures further partition a circle graph, but the fraction of demand routed is not as badly affected (about 0.9 and 0.7 at corresponding points) since only a small amount of demand may have been flowing between the components created after the partition. Similar trends are also observed in targeted attack scenarios - for example ladder graphs under high node betweeness attacks are repoted as far more vulnerable by flow robustness than by fraction routed. Looking at Figure 4, the same applies even when measuring the resilience of real networks - for the NTT network under high utilization link targeting attacks, flow robustness remains at 1.0 for 20 iterations, whereas the fraction of demand routed has dropped drastically from almost 100% to about 60% (i.e. 40% of the demand can no longer be routed) by that time since large amounts of capacity have been lost from the network. This clearly demonstrates that node-pair connectivity measures traditionally used by researchers, may not appropriately capture the resilience of internet backbone networks when the measure of success is

[4] flow robustness $= \frac{\sum_{i=1}^{k} |C_i|*(|C_i-1|)}{|n|*(|n|-1)}$, where C_i is the i^{th} component of the graph (total k) and n is the set of nodes

[5] All the data in the legends are in the figures, if some appear missing it is because the curves overlap exactly - for example in the star graph the highest degree, between centrality and closeness node is the central node and eliminating it makes the fraction routed as well as flow robustness zero in the first iteration in each of these attacks.

the traffic successfully carried. We also note from this that connectivity measures can fail to capture the importance of link failures / attacks, which is probably why the related studies usually ignored them.

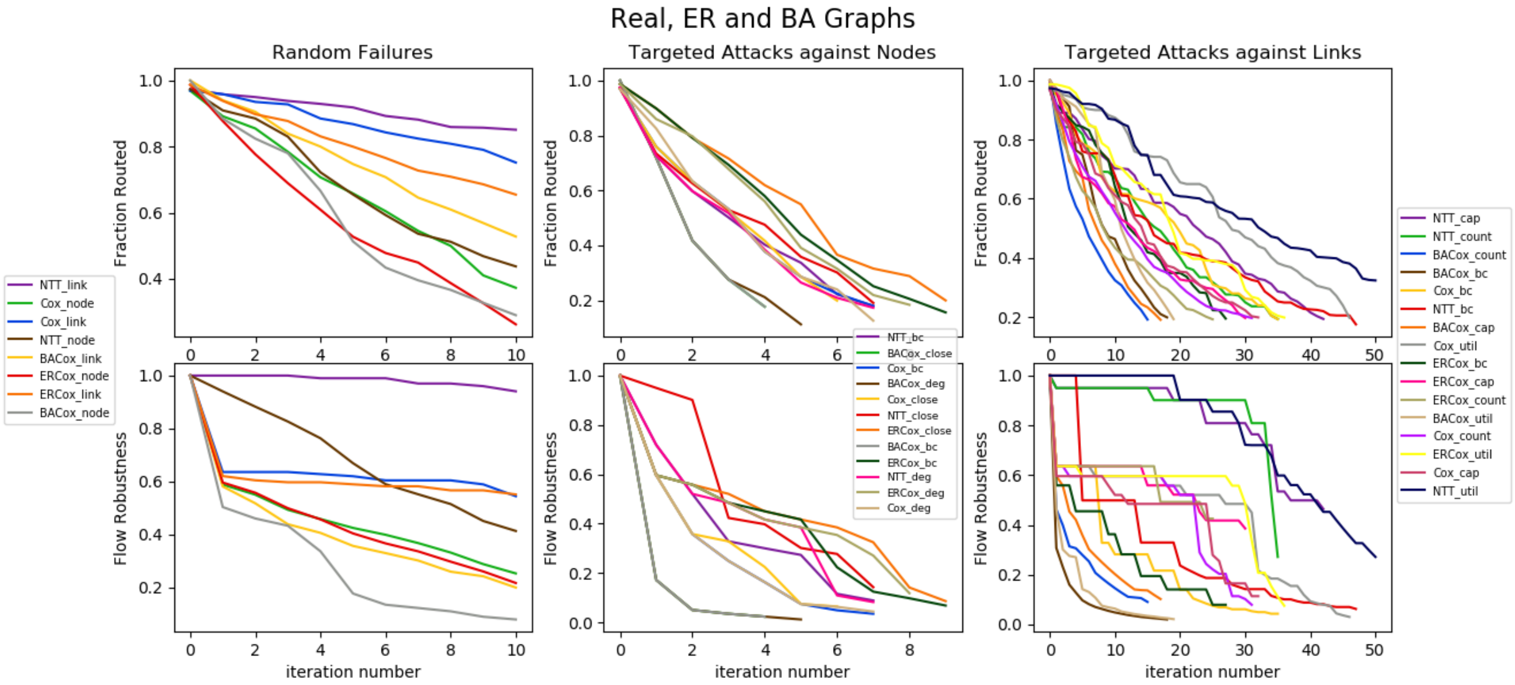


Figure 4: Results for Real and Generated Graphs

5.2.2 Graph Metrics Correlation

In Table 1 we present the 14 graph metrics we consider evaluated for each of the 8 graphs. For ease of presentation, the numbers in this table as well as other tables are in scientific-E notation and rounded to 2 digits (the default float precision in Python was used to run the statistical tests). In Table 2 we present our resilience measure, the sum of fraction routed values (defined in Section 3.3), across iterations for each of the attack and failure types for each of the same 8 graphs. We then compute the Spearman rank correlation coefficient ρ for each column in Table 1 with each column in Table 2, to compute the entries in Table 3. Across each row in Table 3, we highlight the cells that have the highest absolute value, which identifies the metric that was most correlated with our measure of resilience for each attack type.

	AC	NC	EGC	NaC	σ_{NB}	σ_{EB}	μ_{NB}	μ_{EB}	μ_{NDP}	σ_{NDP}	μ_{EDP}	σ_{EDP}	λ_{min}^{CTNC}	λ_{min}^{NDTND}
circle	2.5E-02	2.3E+15	7.3E-03	8.2E-01	0.0E+00	0.0E+00	1.8E+02	4.0E+02	2.0E+00	0.0E+00	2.0E+00	0.0E+00	6.9E-01	7.6E+01
wheel	1.0E+00	-6.6E+14	5.9E-02	3.7E+00	1.1E+04	1.2E+03	1.8E+01	3.8E+01	3.0E+00	0.0E+00	3.0E+00	0.0E+00	3.2E+01	1.1E+02
star	1.0E+00	7.1E+14	2.6E-02	2.6E+00	1.3E+04	0.0E+00	1.9E+01	7.8E+01	1.0E+00	0.0E+00	1.0E+00	0.0E+00	1.0E+01	3.8E+01
ladder	2.5E-02	-1.2E+15	1.3E-02	1.2E+00	3.9E+03	2.1E+04	1.2E+02	2.0E+02	2.0E+00	2.3E-02	2.0E+00	2.3E-02	1.2E+00	7.6E+01
ERCox	3.2E-01	5.0E+14	5.7E-02	1.9E+00	6.5E+02	4.1E+02	2.7E+01	4.5E+01	2.6E+00	1.7E+00	2.6E+00	1.7E+00	9.2E+00	3.0E+01
BACox	7.5E-02	-1.5E+15	1.8E-02	9.8E-01	6.9E+03	1.0E+04	3.9E+01	1.1E+02	1.0E+00	0.0E+00	1.0E+00	0.0E+00	1.7E+00	3.0E+01
Cox	4.9E-01	3.7E+15	5.4E-02	2.4E+00	1.4E+03	5.0E+02	2.5E+01	4.3E+01	1.9E+00	1.1E+00	2.2E+00	1.6E+00	9.6E+00	3.0E+01
NTT	2.8E-01	-1.4E+15	4.2E-02	2.9E+00	3.7E+03	1.9E+03	3.6E+01	5.4E+01	2.0E+00	1.1E+00	2.2E+00	2.1E+00	9.8E+00	3.8E+01

Table 1: Metrics for all 8 graphs

	node_deg	*node_bc*	*node_close*	*link_cap*	*link_bc*	*link_util*	*link_count*	*node_rand*	*link_rand*
circle	8.0E+00	7.1E+00	7.1E+00	1.2E+01	1.2E+01	1.5E+01	1.1E+01	7.5E+00	8.7E+00
wheel	2.2E+00	1.9E+00	1.9E+00	2.1E+01	2.0E+01	3.0E+01	1.5E+01	8.1E+00	9.8E+00
star	1.0E+00	1.0E+00	1.0E+00	6.7E+00	1.8E+01	2.0E+01	6.9E+00	8.2E+00	8.7E+00
ladder	7.6E+00	8.2E+00	8.5E+00	1.7E+01	1.6E+01	2.2E+01	1.3E+01	7.4E+00	9.2E+00
ERCox	5.0E+00	5.3E+00	5.7E+00	1.5E+01	1.5E+01	2.1E+01	1.2E+01	6.4E+00	8.9E+00
BACox	2.7E+00	2.6E+00	2.6E+00	8.9E+00	1.0E+01	1.2E+01	7.6E+00	6.5E+00	8.3E+00
Cox	4.0E+00	4.0E+00	3.8E+00	1.6E+01	1.9E+01	2.9E+01	1.5E+01	7.3E+00	9.6E+00
NTT	3.8E+00	3.9E+00	4.2E+00	2.3E+01	2.2E+01	3.1E+01	1.8E+01	7.5E+00	1.0E+01

Table 2: Sum of Fraction Routed for all 8 graphs

From Table 3 we see that with our measure of resilience, the first 6 metrics from the left of table which were evaluated in [6] based on a different measure of resilience only emerge as the winner 5 times (including a tie) out of 9. We also learn that even if we leave out random node failures, which none of the metrics had a high correlation with, there is not one metric

	AC	NC	EGC	NaC	σ_{NB}	σ_{EB}	μ_{NB}	μ_{EB}	μ_{NDP}	σ_{NDP}	μ_{EDP}	σ_{EDP}	λ_{min}^{CTNC}	λ_{min}^{NDTND}
n_deg	-7.6E-01	2.4E-01	-4.5E-01	-6.7E-01	-8.6E-01	1.2E-02	7.6E-01	4.8E-01	3.6E-01	3.8E-01	7.2E-02	3.0E-01	-8.3E-01	1.6E-01
n_bc	-7.4E-01	1.4E-01	-4.3E-01	-6.2E-01	-7.6E-01	1.7E-01	7.4E-01	4.5E-01	3.8E-01	4.4E-01	9.6E-02	3.7E-01	-8.1E-01	1.6E-01
n_close	-7.9E-01	0.0E+00	-4.5E-01	-5.7E-01	-7.4E-01	2.2E-01	7.9E-01	5.0E-01	4.1E-01	4.2E-01	1.2E-01	4.2E-01	-7.9E-01	2.3E-01
l_cap	9.5E-02	-2.6E-01	4.0E-01	5.2E-01	-1.2E-01	5.0E-01	-9.5E-02	-4.3E-01	5.9E-01	3.9E-01	7.8E-01	5.7E-01	2.6E-01	4.9E-01
l_bc	6.0E-01	4.8E-02	5.7E-01	9.0E-01	2.6E-01	8.4E-02	-6.0E-01	-6.4E-01	1.9E-01	2.4E-01	5.7E-01	4.2E-01	7.6E-01	3.6E-01
l_util	4.3E-01	-4.8E-02	6.0E-01	7.9E-01	4.8E-02	2.8E-01	-4.3E-01	-6.4E-01	4.3E-01	4.4E-01	7.7E-01	6.0E-01	5.7E-01	4.0E-01
l_count	1.9E-01	-1.4E-01	5.0E-01	5.7E-01	-1.7E-01	4.1E-01	-1.9E-01	-5.5E-01	5.1E-01	4.4E-01	8.0E-01	6.0E-01	3.3E-01	3.8E-01
n_rnd	3.8E-01	9.5E-02	0.0E+00	5.7E-01	5.5E-01	-2.5E-01	-3.8E-01	-4.8E-02	-4.8E-02	-5.6E-01	-4.8E-02	-3.6E-01	5.7E-01	6.1E-01
l_rnd	4.3E-01	-4.8E-02	6.0E-01	7.9E-01	4.8E-02	2.8E-01	-4.3E-01	-6.4E-01	4.3E-01	4.4E-01	7.7E-01	6.0E-01	5.7E-01	4.0E-01

Table 3: Spearman Rank Correlations of Graph Metrics to Sum of Fraction Routed

that does well against all the attacks, and multiple metrics are needed to get adequete coverage across attack types - this is especially true for attacks targeting links where barring the highly utilized link targeting attack, in all the other cases the gap in correlation coefficient between the winner and second placed metrics was quite significant.

We observe that attacks targeting nodes are usually highly correlated with metrics that are measuring network node properties - specifically, the variance of node betweeness, σ_{NB} is most highly correlated with the attack that targets high degree nodes, while the mean node betweeneness, μ_{NB}, best captures the attack where high closeness nodes are targeted. Interestingly, although they too are highly correlated with high betweenness node targeted attack's sum of fraction routed, λ_{min}^{CTNC} beats both σ_{NB} and μ_{NB} for this attack - this shows that since the measure of resilience has to do with fraction of demand routed, a metrics that accounts for the network capacities can do better than ones that are directly related to the attack criterion. This suggests that further research towards identifying graph metrics that account for edge weights can be fruitful for measuring telecommunications network resilience.

We see that μ_{EDP}, the mean number of edge disjoint paths, winsl when it comes to attacks that target high capacity links and links that support a high number of demands and also does well for high utilization link targeting attacks. This is not entirely surpising since usually multiple paths of same or similar capacity exist in a network so despite being capacity unaware, the metric does quite well especially since we allow for demand to be split and routed along any available paths in our model. Indeed, usually telecommunication network planners design networks with multiple redundant paths that have similar capacities. For attacks targeting links with high betweenness, high utilizations as well as for random link failures, the natural connectivity, NAC, turns out to be the best metric. Unlike many of the other spectral metrics which pick an eigenvalue of a matrix related to the graph, the NAC, is a scaled average of all of the eignevalues of the adjacency matrix, which seems to suggest that for link attacks, spectral metrics that account for the entire spectral decomposition may be more useful, which can inform future research.

6 Conclusions and Future Work

In this paper, we modeled the telecommunications backbone network as a capacitated graph, and proposed a new measure of their resilience based on the demand carrying capability of the network unlike previous approaches that attempted to evaluate their resilience against random failures and targeted attacks from a network science perspective. We use a fractional multi-commodity flow maximization problem modeled as a linear program to compute our measure of resilience, and ran simulation experiments modeling a total of 9 failure and attack scenarios including random link failures and targeted attacks against links. We discovered that previous approaches that used measures of resilience based on node-pair connectivity may significantly over or under estimate resilience of the network in terms of its demand carrying capability, which in practice is the metric that internet service providers would care about. We then investigated a set of 14 metrics, 8 of them not evaluated in the literature, and identified that several of the previously unevaluated metrics are actually most highly correlated with some attack / failure types.

In terms of future work, from a network science perspective, our results regading correlations between graph metrics and sum of fraction routed were empirical - a more theoretical analysis is conceivable given that as seen in Section 3.2 there is an intimate connection between network flows and path diversity metrics such a number of node / edge disjoint paths which generally did well as graph metrics in our results. We believe that there are likely yet undiscovered graph metrics that account for edge weights which are better suited to identifying network resilience in terms of demand carrying capacity, which is another direction of future work. From a telecommunications networking perspective, our current fractional multi-commodity flow model did not account for traffic quality of service - for example some demands may have to be routed on low latency paths, but we did not impose any restrictions on even the length let alone link capacities of the paths taken by demands in our model. Similarly, some traffic may be sensitive to jitter (latency variance), in which case we cannot split a single flow into how many ever pieces as desired as done in our current model. So our results in terms of fraction of demand routed are really a best-case scenario, and we intend to build a more complex model accounting for such practical issues.

References

[1] Daniel J. Rosenkrantz, Sanjay Goel, S.S. Ravi, and Jagdish Gangolly. Resilience Metrics for Service Oriented Networks: A Service Allocation Approach. IEEE Transactions on Services Computing, Vol. 2, pp. 183-196, 2009.

[2] Wenjun Wang, W. Nick Street,. and Renato E deMatta. Topological Resilience Analysis of Supply Networks under Random Disruptions and Targeted Attacks. Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 250-257, 2015.

[3] K. Zhao, A. Kumar, T. P. Harrison, and J. Yen. Analyzing the resilience of complex supply network topologies against random and targeted disruptions. IEEE Systems Journal, Vol. 5, no. 1, pp. 2839, 2011.

[4] R. Albert, H. Jeong, and A. Barabasi. Error and attack tolerance of complex networks. Nature, Vol. 406, pp. 378–381, 2000.

[5] C. Palmer, G. Siganos, M. Faloutsos, C. Faloutsos, and P. Gibbons. The Connectivity and Fault-Tolerance of the Internet Topology. In Workshop on Network-Related Data Management (NRDM), 2001.

[6] Mohammed J. F. Alenazi, and James P. G. Sterbenz. Comprehensive Comparison and Accuracy of Graph Metrics in Predicting Network Resilience. Proceedings of 2015 11th International Conference on the Design of Reliable Communication Networks (DRCN), pp. 157-164. 2015.

[7] https://en.wikipedia.org/wiki/Laplacian_matrix

[8] http://snap.stanford.edu/snappy/index.html

[9] http://www.geeksforgeeks.org/find-edge-disjoint-paths-two-vertices/

[10] https://lucatrevisan.wordpress.com/2011/02/04/cs261-lecture-9-maximum-flow/

[11] http://jeffe.cs.illinois.edu/teaching/algorithms/notes/24-maxflowapps.pdf

[12] http://pymprog.sourceforge.net/index.html

[13] http://www.cox.com/wcm/en/business/datasheet/national-ip-backbone-map.pdf

[14] http://www.us.ntt.net/about/network-map.cfm

[15] https://en.wikipedia.org/wiki/Optical_Carrier_transmission _rates

[16] https://en.wikipedia.org/wiki/Digital _Signal _1