This paper was presented at RNDM 2014 - 6th International Workshop on Reliable Networks Design and Modeling (Nov. 17-19, 2014, Barcelona, Spain)

Cost-Constrained and Centrality-Balanced Network Design Improvement

Mohammed J.F. Alenazi*‡, Egemen K. Çetinkaya§⋆, and James P.G. Sterbenz*†
*Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS 66045, USA
{malenazi, jpgs}@ittc.ku.edu
‡College of Computer and Information Sciences
Department of Computer Engineering
King Saud University, Riyadh, Saudi Arabia
mjalenazi@ksu.edu.sa
§Department of Electrical & Computer Engineering
Missouri University of Science and Technology, Rolla, MO 65409, USA
cetinkayae@mst.edu
†School of Computing and Communications (SCC) and InfoLab21
Lancaster LA1 4WA, UK
jpgs@comp.lancs.ac.uk
www.ittc.ku.edu/resilinets

***Abstract*—Improving resilience against failures and targeted attacks is an important aspect of network design. The resilience and cost of networks are two opposing objectives in which a designer should consider when building networks. We develop a heuristic algorithm that balances the centrality of networks by adding a set of links that minimizes the variance of graph centrality measures in a least costly fashion. Moreover, our algorithm limits the addition of links by a budget constraint. We apply our algorithm to three different realistic topologies and measure the performance of the improved graphs in terms of flow robustness when subjected to targeted attacks. Our results indicate that degree-balanced networks are more resilient than both betweenness-balanced and closeness-balanced networks.**

***Index Terms*—Network design, optimization, augmentation, algorithm; Network cost model; Network resilience, survivability, connectivity, robustness, dependability, reliability; Centrality metrics, betweenness, closeness, degree**

I. Introduction and Motivation

Network design and optimization has been studied by many researchers in the past decades. From a graph perspective, the objective is to improve the connectivity of a given non-empty or empty graph by adding a set of links that maximizes a given connectivity function. A network cost is associated with the design and optimization of networks in terms of nodes, links, or both. The cost and connectivity of graphs are two opposing objectives that a designer should consider [1], [2]. The robustness of graphs can be measured in terms of a number of graph metrics [3]. In this paper, well-known graph centrality metrics are used to measure the importance of a node in terms of betweenness, closeness, and degree [4]. The reason these centrality metrics are chosen is twofold. First, an adversary with knowledge of the network topology can attack the most central nodes with the intention to cause the most damage [5]. Second, from a load-balancing perspective, the flows are more evenly distributed in centrality-balanced graphs. Therefore, centrality metrics provide a good means of measuring resilience [1] and load-balancing traffic.

In this paper, our assumption is that the cost of wide area networks is dominated by the link length, and thus we ignore the cost associated with nodes as well as any variable costs that allow the cost of a network to be captured in terms of total link length [5]–[7]. We improve networks by using this cost model to add links. In this paper, we present a greedy algorithm that adds links to make a given graph more resilient against targeted attacks. To increase the resilience, we focus on improving the graphs to balance their node centrality. The centrality properties attract adversaries who apply successful attacks on a targeted network by disrupting a few nodes with high centrality. To improve the graph, we add links with an objective function to *minimize the centrality variance* of the nodes, which in turn yields a *centrality-balanced* graph. As a result, the adversary needs more resources and an increased work factor to successfully attack.

The rest of the paper is organized as follows: We present a brief background on network optimization and graph centrality metrics in Section II. The assumptions, objective functions, and our heuristic algorithm are presented in Section III. The dataset for the communication networks as well as evaluation of these topologies using our algorithm are presented in Section IV. Finally, we summarize our findings and propose future work in Section V.

⋆ Work performed while at The University of Kansas.

978-1-4799-7039-1

II. Background and Related Work

Network design and optimization has been studied in the past decades [8], [9] and many problems in this field are considered to be NP-hard [10]–[13]. Moreover, adding a set of links or nodes to graphs to optimally maximize a certain graph property is known in the literature as *graph augmentation* and also proven to be NP-hard [14], even for one objective function such as optimally increasing the algebraic connectivity [15]. Next, we briefly present some of the recent work relevant to ours. Algorithms have been developed to add links between a random pair of nodes and two low-degree nodes to improve the connectivity of graphs [16]. The largest connected component has been measured as an objective function on synthetically generated graphs [16]. Another algorithm to improve the robustness of the networks was presented in which some links were rewired [6]. Most relevant to our work in this paper is an algorithm that minimizes the maximum node betweenness of a graph applied to synthetically generated Erdős-Rényi and Barabási-Albert graphs [17], [18]. In addition to these algorithms, we developed a heuristic algorithm that improves the connectivity of a graph using the algebraic connectivity metric by adding links in a cost-efficient fashion in our earlier work [7], [19].

A plethora of graph metrics exist in the literature [3]. There are also several centrality metrics; however, we focus on the best known ones: betweenness, closeness, and degree centrality [20], [21]. Degree centrality is the number of links incident to a node and can be viewed as the importance of connectivity of a node [4]. Betweenness is defined as the number of the shortest paths that flow through a node; it signifies a node's importance in communication [22]. Closeness is the inverse of the sum of the shortest paths from a node to every other node and indicates efficiency of a message's diffusion in a network [4]. While degree centrality provides local information about a node's significance, the betweenness and closeness centrality metrics provide global information about a node's significance. These graph centrality metrics have been used to study performance of networks against targeted attacks [23], [24].

III. Improvement Algorithm

In this section, we describe our algorithm that balances graph centrality by minimizing the centrality variance of a given graph based on a given centrality function. The centrality functions used in this paper are node betweenness, node closeness, and node degree.

A. Algorithm

The objective of this algorithm is to balance graph centrality among all the nodes of a given graph by adding a set of links constrained by a cost budget. To achieve this objective, our algorithm minimizes the variance of the node centralities measured by one of the three node centrality functions: node betweenness, node closeness, and node degree. If there are multiple links that yield the same minimum variance value, the lowest cost link is selected. The pseudocode of our algorithm is shown in Algorithm 1.

Functions:
cost(l) := cost of link l
nBtwn(G) := betweenness for all nodes in graph G
nClos(G) := closeness for all nodes in graph G
nDeg(G) := degree for all nodes in graph G
candidate(G) := candidate links function
var(L) := computes variance of list L
bestLink(L) := affordable low variance in list L
Input:
G := input graph
B := available budget
Output:
selectedLinks := an ordered list of the selected links
begin
  centralityFunc = nBtwn | nClos | nDeg
  selectedLinks = [] ; empty ordered list
  varAndCost = [] ; empty ordered list
  totalCost = 0; initial total cost is zero
  while $B >= \text{totalCost} \wedge$
  $\text{selectedLinks} \neq \text{candidate}(G)$ **do**
    G.addlinks(selectedLinks)
    for l in $\text{candidate}(G)$ **do**
      centralityVar = var(centralityFunc(G))
      varAndCost.append((l, centralityVar, cost(l)))
    end
    selectedLink = bestLink(varAndCost)
    selectedLinks.add(selectedLink)
    totalCost += cost(selectedLink)
  end
  return selectedLinks
end

Algorithm 1: Balancing centrality algorithm

There are two inputs for this algorithm: an input graph G and a budget constraint B. The input graph G has a number of nodes n with a number of links l and the node positions. The budget constraint B is measured in meters to specify the allowed total length for link addition. The algorithm adds links to the graph iteratively. To keep track of the selected links in each iteration, the algorithm adds these links to the selectedLinks list. Moreover, to keep track of the selected links cost, the algorithm increments the totalCost by the cost of each added link.

For the candidate set, all possible candidate links are in the graph's complement. However, this set may contain very long links that are not practical to be added to a physical graph. For example, adding a physical fiber link between Los Angeles and Boston is unlikely to be feasible for providers given the high cost incorporated by adding this link. Therefore, this raises the question of what the best threshold value should be. In this paper, we choose the maximum link length in the input graph to be the threshold for removing long links from the graph complement links. We assume this value gives a good indicator for the maximum link length a provider can afford.

There are seven functions used by this algorithm. The cost function cost(l) returns the cost of adding a link l that is defined as the Euclidean distance between the two ends of the link. The function nBtwn(G) computes the node betweenness of all the nodes in graph G. The function nClos(G) computes the closeness for every node in graph G. The function nDeg(G) computes the number of links connected to every node in graph G. The bestLink(L) function returns the best candidate link given that it is an affordable link and it has the minimum-variance value and lowest cost in case of multiple tie minimum-variance values. The var(L) function returns the variance of the values in list L. The candidate(G) function returns candidate links in graph G.

The centrality function centralityFunc is selected from the three options: nBtwn, nClos, and nDeg. For each link in the candidate set, the algorithm temporarily adds the link to the graph and computes the variance value and the cost incurred by that link, which are added to the varAndCost list. After that, the temporary link is removed and the next candidate link is added to undergo the same process. The link with the minimum variance and the lowest cost is selected using the bestLink function and then it is added to the selectedLinks list. Inside the bestLink function, if the minimum variance link cost plus the total cost exceeds the budget, a next minimum link is examined until a link with affordable cost is found. This process is repeated until no link can be added without exceeding the given budget or there are no more available links in the candidate set.

B. Improvement Example

In this section, we explain how our heuristic algorithm balances the centrality of a small-size graph. Figure 1 shows a sample graph connecting major US cities with 7 nodes and 7 links. In this example, we apply our algorithm to add a single link using three objective functions one at a time.

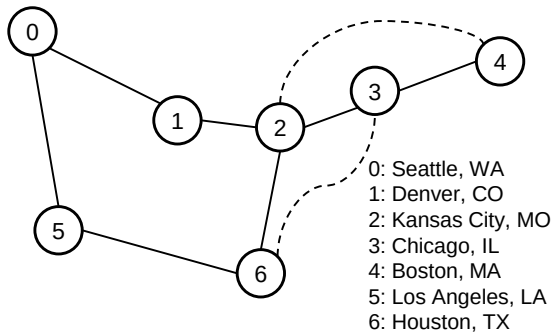


Fig. 1. Improvement graph example

To find the best candidate, the algorithm finds the candidate set, which contains the links of the complement graph that are not longer than the current maximum link in the graph. The number of links in this complement graph is $\frac{7\times6}{2} - 7 = 14$ links. In this example, the longest link of the input graph is between nodes 5 and 6 that has a length of 2,177 km.

TABLE I
CENTRALITY VARIANCE AND COST FOR CANDIDATE LINKS

Candidate links	Betweenness variance	Closeness variance	Degree variance	Cost [m]
(1, 3)	0.0125	0.0085	0.4897	1,453,452
(1, 5)	0.0374	0.0086	0.4897	1,259,832
(1, 6)	0.0323	0.0099	0.4897	1,070,221
(2, 4)	0.0379	**0.0076**	0.4897	1,988,059
(2, 5)	0.0459	0.0115	0.7755	2,143,391
(3, 6)	**0.0125**	0.0085	**0.4897**	**1,043,873**

Therefore, 8 links that have a length greater than this value are removed from the candidate link set and 6 links remain in the candidate set. For each candidate link, the algorithm determines the centrality variance across all nodes after adding the candidate link and the cost incurred as shown in Table I. According to this algorithm, the link resulting in the minimum centrality variance is selected. For example, using closeness variance as an objective function, the link (2,4) is selected because it gives the minimum variance value of 0.0076. In case there are multiple identical minimum values, the link with the least cost is selected among those minimum variance links. For example, using betweenness variance as an objective function, there are two links with the minimum value of 0.0125, namely links (1,3) and (3,6). Next, the algorithm chooses the link with the minimum cost, which is (3,6) in this example.

IV. Results and Analysis

In this section, first we present the topological data. Next, we apply the optimization algorithm on three realistic networks and study the results. Then, we apply three centrality-based attacks to the resulting improved graphs and show how the robustness changes during each attack.

A. Topological Dataset

We study physical infrastructure communication networks that are geographically located within the continental United States. Therefore, we only include the 48 contiguous US states, the District of Columbia, and exclude Hawaii, Alaska, and other US territories. We make use of the publicly available Internet2 [25], Level 3 [26], and CORONET [27], [28], fiber-level topologies. Important graph metrics for these physical-level topologies are shown in Table II and a detailed analysis of graph metrics for the given physical networks was presented in our earlier work [5]. Next, we apply our improvement algorithm on the Internet2, CORONET, and Level 3 fiber-level topologies.

B. Improvement Analysis

In this section, we apply our improvement algorithm on three service provider graphs and study the improvement and the cost incurred for each graph as links are added. We choose an upper limit of the budget constraint to be 5×10^7 meters, such that the number of added links across the three different size physical-level graphs demonstrate distinctive centrality improvement. The budget is measured as the sum of the length

TABLE II
TOPOLOGICAL CHARACTERISTICS OF PHYSICAL-LEVEL TOPOLOGIES

Network	Nodes	Links	Node Degree (min. / avg. / max.)	Diameter	Radius	Hopcount	Node Closeness (min. / avg. / max.)	Node Betweenness (min. / avg. / max.)
Internet2	57	65	(2.0 / 2.28 / 4.0)	14	8	6.69	(0.12 / 0.15 / 0.21)	(15 / 159 / 630)
CORONET	75	99	(2.0 / 2.64 / 5.0)	17	9	6.45	(0.10 / 0.16 / 0.22)	(6 / 201 / 1,090)
Level 3	99	130	(1.0 / 2.67 / 5.0)	19	10	7.65	(0.09 / 0.14 / 0.18)	(0 / 325 / 1,622)

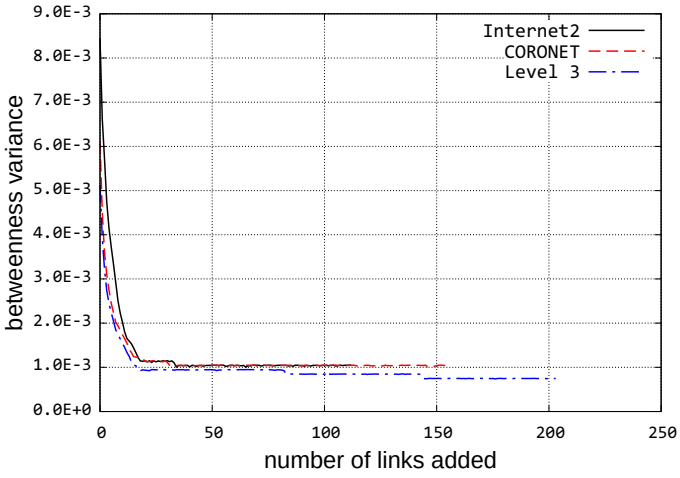
of the added links. We note that this value can be modified according to a provider's available budget. The results for all of the objective functions are shown in Figure 2.

1) Balanced Betweenness Analysis: We apply our algorithm to the three physical graphs while the objective function is set to minimize betweenness. In other words, links are added to uniformly utilize all nodes in terms of forwarding traffic along shortest paths. The variance changes while links are added is shown in Figure 2a. For all providers, the betweenness variance starts around 7×10^{-3}. While adding the first 20 links the variance decreases significantly to 1×10^{-3}; these links contribute significantly to minimizing the variance and the remaining links do not have any significant impact. Therefore, the variance slowly decreases as the rest of the links are added because the graphs have been already balanced in terms of their betweenness values variance. The costs incurred as links are added for all providers are shown in Figure 2b. The cost is increasing on a similar pace for all providers but their slopes are different. The cost of adding the first 20 links is growing faster than afterwards because these links are selected regardless of their expensive cost since they contribute significantly to minimizing the betweenness. Level 3 has the lowest slope, which gives the highest number of added links. This is because Level 3 has the largest number of nodes, which gives more affordable candidate links to select from. On the other hand, the incurred cost while adding links to Internet2 grows faster than the other two because it has a lower number of nodes, which in turn yields a lower number of candidate links. As a result, more expensive links are selected, which consumes the budget more quickly.

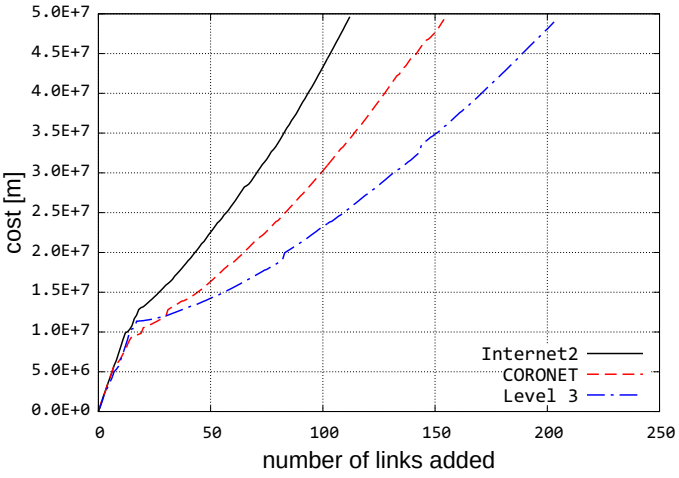
2) Balanced Closeness Analysis: While selecting the objective function to minimize the variance of the node closeness of the graph, we apply our algorithm to the three graphs. In other words, links are added to make the shortest path distance between all the nodes more uniform. The variance changes while links are added is shown in Figure 2c. The variance of closeness values are different for the three providers. For Level 3 the closeness variance decreases overall. On the other hand, for Internet2 and CORONET, the variance decreases while adding the first several links and then it fluctuates around 5.5×10^{-4}. However, why the fluctuations happen *only* for closeness-based optimization is not known and the reasons for the occurrence of this phenomenon will be the subject of future work. The costs incurred as links are added for all providers are shown in Figure 2d. Here, the costs are similar with no phase changes since the added links do not have a significant contribution to minimize closeness variance for Internet2 and CORONET. For Level 3, the cost exhibits the same slope of the other two providers overall. However, while adding the 60th link, we observe a small jump in the cost, which corresponds to a significant decrease in the variance for the same link. This is because the algorithm selects links that minimize the variance regardless of their cost, which in this case is higher than other selected links.

3) Balanced Degree Analysis: Here, we apply our algorithm to the three physical graphs while the objective function is set to minimize degree variance. In other words, links are added to increase the uniformity of node degree. The variance changes while links are added as shown in Figure 2e. For all the providers, the degree variance starts from different initial values but they are *not monotonically decreasing, but rather oscillating.* To explain this phenomenon, let us start with a uniform node degree graph, where each node has a degree of k, which gives a graph with zero degree variance. To add links to the graph, the variance has to increase no matter where the links are placed. The variance increases until it reaches a top point and then it decreases to reach the zero where the graph has a $k+1$ node degree. If the number of links needed to increase k to $k+1$ is x, then the top points must be located near adding the next $x/2$ links. Now, we can observe that the degree variance in Figure 2e does not reach zero for any providers. This is because long links can not be selected due to the cost constraints. The costs incurred as links are added for all providers are shown in Figure 2f. For all providers, the cost increases overall with a similar pattern of phase changes. For example, Level 3 link addition cost increases at the same rate. Then, the rate increases until the 27th link is added, which slows the rate of cost increase. The point where the cost slows down corresponds to the point where the degree variance is at a minimum. This happens because at this point, the graph has the maximum number of candidate links to get the *next* minimum degree variance of the graph. However, after adding a few links with the low cost, the remaining links in the candidate set are all expensive. Therefore, the algorithm has to select one of these links, and the links added before reaching the lowest variance are more expensive than links selected after passing this point.

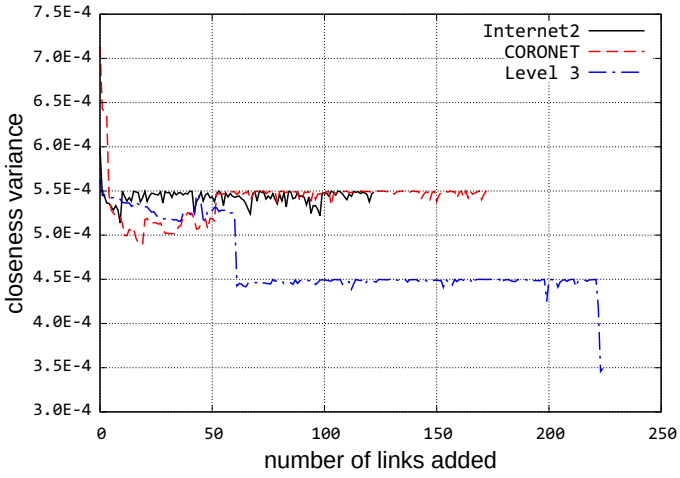
4) Improvement Method vs. Number of Added links: Using the improvement results shown in Figure 2, we observe that while limiting the budget to a constant value for all the graphs, the actual number of added links for a given graph differ based on the used improvement method. For example,



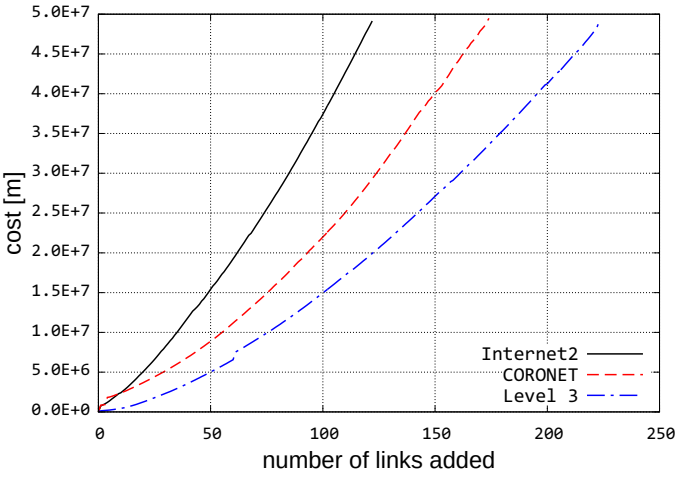
(a) Minimizing variance of node betweenness



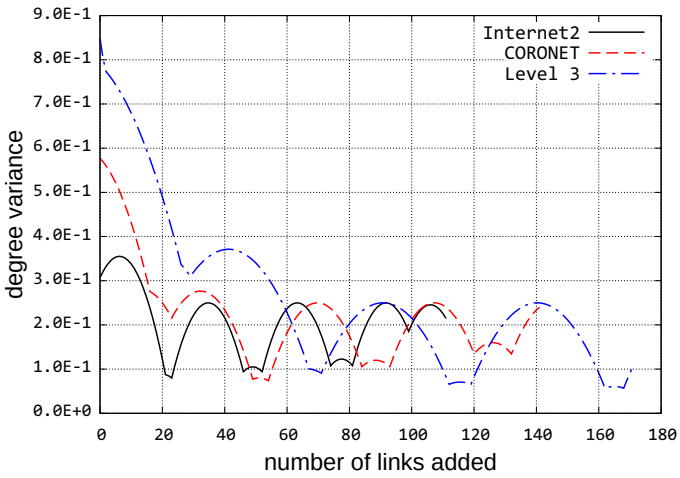
(b) Cost of balancing node betweenness



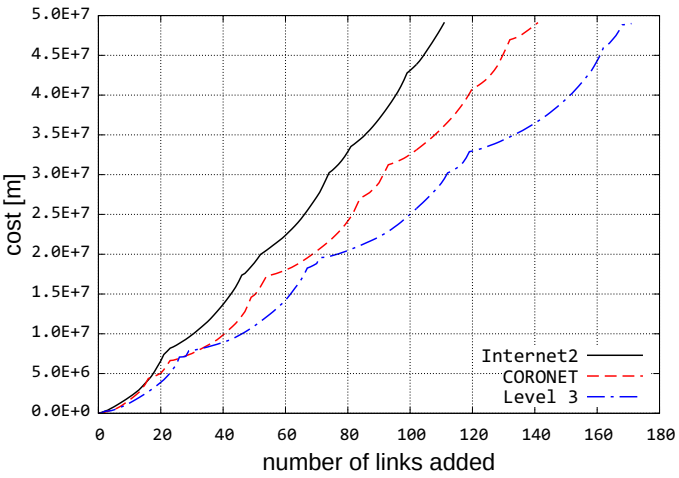
(c) Minimizing variance of node closeness



(d) Cost of balancing node closeness



(e) Minimizing variance of node degree



(f) Cost of balancing node degree

Fig. 2. Optimizing physical-level topologies with 50M budget constraint

for the Internet2 graph, the numbers of added links using betweenness, closeness, and degree are: 112, 122, and 112, respectively. For the CORONET graph, the numbers of added links using betweenness, closeness, and degree are: 154, 174, and 141, respectively. For the Level 3 graph, the number of added links using betweenness, closeness, and degree are: 203, 224, and 171, respectively. From these numbers, we observe that closeness improvement methods always yield the highest number of added links, which implies that it tends to select shorter links. On the other hand, both betweenness and degree improvement yield a fewer number of links. The Degree-based improvement method yields the lowest number of links added with small differences with respect to betweenness-based improvement.

C. Robustness Evaluation

In this section, we present the set of attacks used to evaluate the robustness of the resulting non- and improved graphs. Then, we apply these attacks and show the results.

1) Flow Robustness: Flow robustness [29], [30] is a graph metric that measures the ratio of possible number of pair-connections, to the maximum number of pair-connections $n(n-1)$. If the graph is partitioned, the possible number of pair-connections is the sum of $n(n-1)$ connections for each component. The range of flow robustness value is [0,1], with the flow robustness is 1 if the graph is not partitioned and 0 if the graph has no links.

2) Graph Centrality Attacks: We use a graph theoretic model to attack a given graph and show how its flow robustness changes after each node removal. In this paper, we use three centrality metrics: node betweenness, node closeness, and node degree. Thus, we have three attack models, in which the node with the highest centrality is removed. The node betweenness attack targets the node through which the highest number of shortest paths pass. The node closeness attack targets the closest node to all the other nodes in terms of hop count. The highest degree node attack targets the node with the highest number of neighbors. The list of removed nodes is determined *adaptively* for each attack model. This means the node centrality values are calculated after each node is removed and the highest is selected to be the next node to be removed. The adaptive removal of nodes gives a more correct selection for the highest centrality than the non-adaptive removal, in which the highest targeted number of nodes are selected based on a single evaluation [23].

3) Robustness Evaluation Results: In this section, we show the results of applying the graph centrality attacks to non- and improved graphs while computing the flow robustness of the graph as nodes are removed during the attack, which causes the removal of 50 nodes from each graph. The results of applying the attacks on non- and improved graphs are depicted in Figure 3. The sum of flow robustness values (i.e. the area under each curve in Figure 3) are shown in Table III. By comparing the sum of flow robustness values of the non- and improved graphs, we can see that the betweenness attack overall yields lower flow robustness values, which implies that the betweenness attack is the most destructive attack on these physical graphs. Using the same approach, the closeness attack is next, and the degree attack is the least destructive attack.

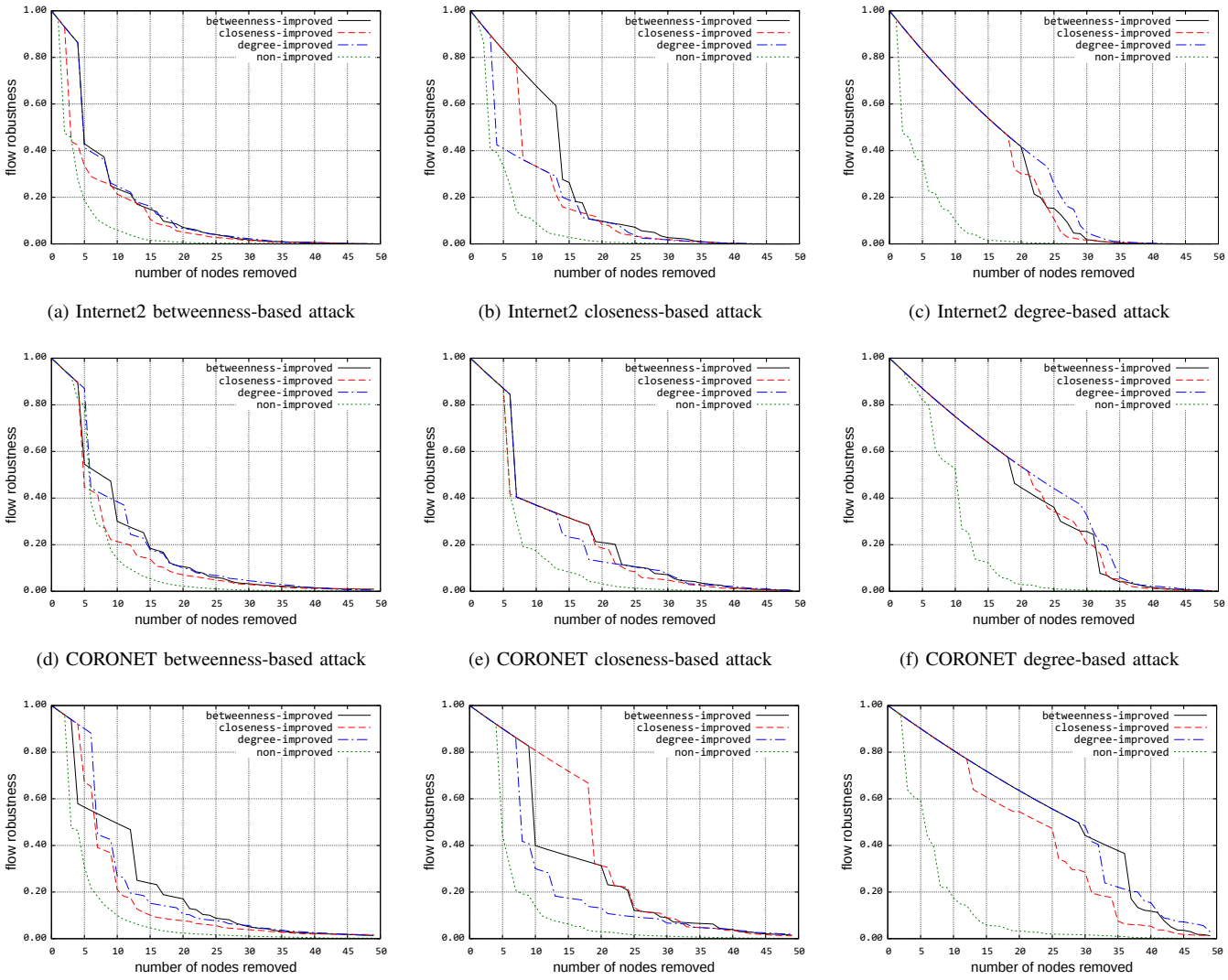
TABLE III
SUM OF FLOW ROBUSTNESS

Provider	Improvement method	Betweenness attack	Closeness attack	Degree attack
Internet2	non-improved	4.09	5.00	4.71
	betweenness	8.65	12.99	15.88
	closeness	6.96	10.28	15.48
	degree	8.68	8.86	16.95
CORONET	non-improved	7.43	7.84	9.87
	betweenness	10.43	12.55	19.72
	closeness	8.76	11.66	20.03
	degree	10.60	11.79	21.28
Level 3	non-improved	5.68	8.86	16.95
	betweenness	11.63	15.36	25.81
	closeness	9.56	18.71	21.54
	degree	11.08	12.07	25.62

The results of applying three centrality attacks on Internet2 non- and improved graphs are shown in Figures 3a, 3b, 3c. For the betweenness attack on Internet2 non- and improved graphs, we observe that the degree-improved graph has the highest value of flow robustness at 8.68. Furthermore, the betweenness-improved graph comes second in terms of flow robustness with very small difference at 8.65. Even though the closeness-improved graph has more added links, it yields the lowest flow robustness among the improvement methods for the Internet2 graph as shown in Table III. For the closeness attack, the betweenness-improved graph outperforms the other methods with flow robustness of 12.99, while closeness and degree flow robustness are 10.28 and 8.86, respectively. For the degree attack, the degree-improved graph again has the highest flow robustness of 16.95. The betweenness and closeness improvement methods come next with flow robustness values of 15.88 and 15.48 respectively. By observing all the flow robustness values for the scenarios we study, the degree improvement is more resilient to attacks for Internet2, and the closeness improvement is the weakest method for the same graph.

Using the same method for analyzing Internet2 flow robustness values, we study both CORONET and Level 3 non- and improved graphs. We observe that the degree improved graphs have the highest flow robustness compared to the other approaches. Next, the betweenness improved graphs have the second highest flow robustness while closeness has the worst flow robustness values.

Finally, we compare the results of the three improvement methods' flow robustness values against the number of added links shown in Figure 2 and discussed in Section IV-B4. We observe that even though the closeness-based improvement consistently yields the highest number of added links, it fails to provide better flow robustness values than both betweenness and degree improvement methods in most attacks. This implies that having a larger number of links in a given graph does



(a) Internet2 betweenness-based attack (b) Internet2 closeness-based attack (c) Internet2 degree-based attack

(d) CORONET betweenness-based attack (e) CORONET closeness-based attack (f) CORONET degree-based attack

(g) Level 3 betweenness-based attack (h) Level 3 closeness-based attack (i) Level 3 degree-based attack

Fig. 3. Flow robustness analysis of non- and improved topologies with 50M meters budget constraint

not necessarily guarantee a better resilience. Moreover, adding links without a careful improvement of networks may not yield any gain in terms of resilience and performance.

V. Conclusions and Future Work

Network design and optimization is a major area of research. Here, we study the improvement of several real-world providers' physical-level graphs by adding links that cost below a certain budget. We introduce an algorithm that minimizes the node centrality variance of a given graph based on three centrality functions: node betweenness, node closeness, and node degree. Then, we apply this algorithm using each function on three physical graphs and study the variance minimization and cost incurred while adding the links. Then, we study resilience of the non- and improved graphs using the centrality attacks via the same centrality functions. For each attack, we study the flow robustness of each non- and improved graph. Overall, the results show the degree-improved graphs outperform the other two improvement methods for these physical graphs. Then, with a similar outcome, the betweenness comes next, and the weakest improvement method is closeness minimization.

Acknowledgments

We would like to acknowledge Dongsheng Zhang for discussions on this work. This research was supported in part by NSF grant CNS-1219028 (Resilient Network Design for Massive Failures and Attacks) and by NSF grant CNS-1050226 (Multilayer Network Resilience Analysis and Experimentation on GENI). Mohammed J.F. Alenazi is supported by SACM (Saudi Arabian Cultural Mission).

REFERENCES

[1] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[2] J. P. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, Q. Shi, and J. P. Rohrer, "Evaluation of Network Resilience, Survivability, and Disruption Tolerance: Analysis, Topology Generation, Simulation, and Experimentation (invited paper)," *Telecommunication Systems*, vol. 52, no. 2, pp. 705–736, 2013.

[3] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167–242, 2007.

[4] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978–1979.

[5] E. K. Çetinkaya, M. J. F. Alenazi, A. M. Peck, J. P. Rohrer, and J. P. G. Sterbenz, "Multilevel Resilience Analysis of Transportation and Communication Networks," *Springer Telecommunication Systems Journal*, 2013. (accepted in July 2013).

[6] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 10, pp. 3838–3841, 2011.

[7] M. J. F. Alenazi, E. K. Çetinkaya, and J. P. G. Sterbenz, "Network Design and Optimisation Based on Cost and Algebraic Connectivity," in *Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, (Almaty), pp. 193–200, September 2013.

[8] R. S. Wilkov, "Analysis and design of reliable computer networks," *IEEE Transactions on Communications*, vol. 20, no. 3, pp. 660–678, 1972.

[9] H. Frank and W. Chou, "Topological optimization of computer networks," *Proceedings of the IEEE*, vol. 60, no. 11, pp. 1385–1397, 1972.

[10] M. O. Ball, "Complexity of network reliability computations," *Networks*, vol. 10, no. 2, pp. 153–165, 1980.

[11] M. O. Ball, "Computational complexity of network reliability analysis: An overview," *IEEE Transactions on Reliability*, vol. 35, no. 3, pp. 230–239, 1986.

[12] S. Khuller and B. Raghavachari, "Graph and Network Algorithms," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 43–45, 1996.

[13] H. Noltemeier, H.-C. Wirth, and S. O. Krumke, "Network Design and Improvement," *ACM Comput. Surv.*, vol. 31, no. 3es, pp. 1–5, 1999.

[14] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 653–665, 1976.

[15] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is NP-hard," *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, 2008.

[16] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish, "Improving network robustness by edge modification," *Physica A: Statistical Mechanics and its Applications*, vol. 357, no. 3–4, pp. 593–612, 2005.

[17] B. Danila, Y. Yu, J. A. Marsh, and K. E. Bassler, "Optimal transport on complex networks," *Phys. Rev. E*, vol. 74, p. 046106, Oct 2006.

[18] B. Danila, Y. Yu, J. A. Marsh, and K. E. Bassler, "Transport optimization on complex networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, 2007.

[19] M. J. Alenazi, E. K. Çetinkaya, and J. P. Sterbenz, "Cost-efficient algebraic connectivity optimisation of backbone networks," *Optical Switching and Networking*, vol. 14, Part 2, pp. 107 – 116, 2014.

[20] S. P. Borgatti, "Centrality and network flow," *Social Networks*, vol. 27, no. 1, pp. 55–71, 2005.

[21] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466–484, 2006.

[22] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[23] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E*, vol. 65, p. 056109, May 2002.

[24] E. K. Çetinkaya, D. Broyles, A. Dandekar, S. Srinivasan, and J. P. G. Sterbenz, "Modelling Communication Network Challenges for Future Internet Resilience, Survivability, and Disruption Tolerance: A Simulation-Based Approach," *Telecommunication Systems*, vol. 52, no. 2, pp. 751–766, 2013.

[25] "Internet2." http://www.internet2.edu.

[26] "Level 3 network map." http://maps.level3.com.

[27] G. Clapp, R. A. Skoog, A. C. Von Lehmen, and B. Wilson, "Management of Switched Systems at 100 Tbps: the DARPA CORONET Program," in *International Conference on Photonics in Switching (PS)*, (Pisa), pp. 1–4, September 2009.

[28] "The Next Generation Core Optical Networks (CORONET)." http://www.darpa.mil/Our_Work/STO/Programs/Dynamic_Multi-Terabit_Core_Optical_Networks_(CORONET).aspx.

[29] J. P. Rohrer, A. Jabbar, and J. P. Sterbenz, "Path Diversification for Future Internet End-to-End Resilience and Survivability," *Springer Telecommunication Systems*, vol. 56, pp. 49–67, May 2014.

[30] J. P. Rohrer, A. Jabbar, and J. P. G. Sterbenz, "Path diversification: A multipath resilience mechanism," in *Proceedings of the IEEE 7th International Workshop on the Design of Reliable Communication Networks (DRCN)*, (Washington, DC), pp. 343–351, October 2009.