A Graph-theoretic QoS-aware Vulnerability Assessment for Network Topologies

Ying Xuan, Yilin Shen, My T. Thai
Department of Computer Information Science and Engineering
University of Florida
Gainesville, Florida, 32611
Email: {yxuan,yshen,mythai}@cise.ufl.edu

Abstract—How to assess the topology vulnerability of a network has attracted more and more attentions recently. Due to the rapid growing number of real-time internet applications developed since the last decade, the discovery of topology weakness related to its quality of service (QoS) is of more interest. In this paper, we provide a novel QoS-aware measurement for assessing the vulnerability of general network topologies. Specifically, we evaluate the vulnerability by detecting the minimum number of link failures that decrease the satisfactory level of the QoS-Optimal source-destination path to a given value, which means a topology with a smaller amount of such link failures is more vulnerable. We formulate this process as a graph optimization problem called QoSCE and provide several exact and heuristic algorithms for various QoS constraint amounts. To our best knowledge, this is the first graph-theoretical framework to evaluate QoS-aware topology vulnerability. Through extensive simulations, the performance of the proposed algorithms is validated in terms of assessment accuracy and time complexity.

I. INTRODUCTION AND RELATED WORK

Network topology vulnerability attracts more and more attentions since the last decade. As summarized in [12], numerous evaluation metrics have been proposed for this purpose, most of which are related with the network connectivity, specifically, how fragmented the network it is in the presence of failures. However, to our best knowledge, none of them take the network quality of service(QoS) into consideration. In this paper, we observe that even before the network being fragmented into pieces, its QoS may already be compromised. In this cases, QoSCE may already drop to an intolerant low level, and the network can no longer provide services. To this end, we present a novel QoS-aware vulnerability assessment framework.

QoS-aware topology vulnerability is critical for the Internet. As the Internet serves as the main carrier of more and more real-time applications, it has to satisfy several QoS measures with predefined thresholds, which include jitter, delay, bandwidth, packet loss and etc. Plenty of QoS routing protocols, e.g., Q-OSPF and PNNI [1], have been developed to meet these requirements. In practical networks, malfunctions often take place at intermediate network nodes/links for routing, consequently, even the optimal routing path from source to destination can satisfy few of the QoS constraints. In this cases, only improvements over routing protocols cannot enhance the robustness in unreliable network environments. Therefore, we are interested in study how many node/link failures are required to break down the network to such an extent. Notice that even the subproblem of this study, detecting an optimal routing path satisfying a set of QoS constraints, is nontrivial.

In practical applications, the constraints that are satisfied by the QoS optimal routing path can be categorized into additive and non-additive ones. Specifically, jitter, delay and packet loss of a routing path are the sum of each metric over all the links belonging to this path. However, constraints like bandwidth are not additive from edge to edge, but min/max or multiplicative functions. In principle, multiplicative measures can be converted into additive ones in a logarithmic manner and min/max non-additive measures can be satisfied by ruling out all the unsatisfied single links. Therefore, classic theoretical studies over the QoS routing are normally formulated as a multi-additive-constraint path (MCP) problem [4][2][3]: Consider a network $G(V,E,s,t)$ with designated source node $s$, destination node $t$, and $m$ additive constraint $(c_1,\cdots,c_m)$, where each edge $(u,v)\in E$ has $m$ additive weights $w_i(u,v)\geq 0,$ $i\in [1,\ldots,m]$. Find a path $P$ from $s$ to $t$ with

$$w_i(P) \triangleq \sum_{(u,v)\in P} w_i(u,v) \leq C_i$$

for all $i\in [1,m]$, if it exists.

MCP has been shown to be NP-complete but not strongly NP-complete [2], thus it is tractable for practical network sizes. Xue et.al [11] proposed approximation algorithms towards this problem and several other variants, however since MCP is merely a subproblem of our assessment, to decide if a given set of node/link is a feasible solution to our problem, the approximation ratio to MCP cannot be grafted. As summarized by Khadivi et al. in [5], plenty of mixed-metric based heuristics have been proposed to tackle MCP, however, any inaccuracies brought by the heuristics will possible provide a solution set that is far from optimal to our problem. In [6], Li et al. modified classic dynamic programming algorithm for MCP and provided a fast exact solution by effectively compacting the search space. However, as to be further defined later, the paths studied in our problem are not limited to those satisfying all given constraints, but may satisfy only a subset of them. Therefore, it is much more challenging than MCP.

The purpose of vulnerability assessment is to discover the weakness of the object network topology, whose result can be applied to optimizing network topology design, enhancing network robustness or destroying terrorist networks. We refer
to these weak nodes/links as critical nodes/links, specifically, the minimum set of nodes/links whose failure can bring down the network QoS to a certain low level are called QoS critical node/link set. Therefore, given two networks of the same size, the one which has a smaller QoS critical node/link set is of course more vulnerable. In this paper, we measure the network QoS by the optimal QoS source-destination routing path, which satisfy the most QoS constraints over all routing paths. By requiring such an optimal path satisfies the multiple QoS constraints, we put a threshold on the network QoS and discover the QoS critical node/link set correspondingly.

Our contributions in this paper are: (1) provide the first graph-theoretic QoS-ware vulnerability assessment method; (2) abstract the assessment problem as a graph optimization problem and study its hardness; (3) present one exact algorithm for practical network settings and one fast heuristic algorithm for general network cases.

The rest of this paper is organized as follows. Notations and the problem model are included in Section II, where an integer program and preliminary hardness discussions are also presented. An exact solution is provided in Section III and an efficient betweenness-based heuristic is introduced in Section IV. Extensive simulation studies over the performance of the proposed assessment algorithms are presented in Section V. Section VI includes some further discussions and concludes the whole paper.

II. PROBLEM MODEL AND HARDNESS

A. Models

The system definition of this problem is: given a network with several additive QoS measures and a designated source-destination pair, there exists a set of source-destination (s-t) paths, each of which satisfies the constraints of some of these measures. Each measure is assumed with different credits (for paths, each of which satisfies the constraints of some of these constraints a satisfactory score, an significant measure will earn a greatest credit for the source-time applications), then satisfying the constraint on the most significant measure will earn a greatest credit for the source-destination path. Therefore, the level of satisfying multiple constraints by each path is quantified as a credit, which we call satisfactory score of the specific path. With a threshold on this satisfactory score, an s-t path is called QoS operational. A network fails if QoS operational s-t path can be found. Henceforth, by checking how many node/link failures a given network topology can tolerate before it fails.

This evaluation process can be abstracted into a graph optimization problem called QoS-Critical Vertices (QoSCV) / QoS-Critical Edges(QoSCCE): Given a directed graph G(V, E, s, t) with m-dim edge weight vector (u, v) ∈ E: (w1(u, v), w2(u, v), ..., wm(u, v)). The weight vector for each s-t path P is defined as (w1(P), w2(P), ..., wm(P)) where w1(P) = \sum_{i=1}^{m} w_i(u, v) for all i ∈ [1, m].

Given a constraint threshold vector (c1, c2, ..., cm) with corresponding credit vector (1, 1, ..., 1), an s-t path P satisfies the i-th constraint (denoted as p ∝ i) iff w_i(P) ≤ c_i, and an SAT score \phi(P) is defined as: \phi(P) = \sum_{j: p ∝ i} \lambda_j.

\min \sum_{e \in E} (1 - X_\epsilon) \\
\text{s.t.} \\
\sum_{e \in P} Y_{Pi} \lambda_\epsilon \leq \rho, \ \forall s-t \ path \ P \\
\sum_{e \in P} w_i(e) \leq \epsilon - Y_{Pi} + \sum_{e \in P} (1 - X_\epsilon) \epsilon + c_i, \ \forall i, P \\
\sum_{e \in P} w_i(e) > 1 - Y_{Pi} \epsilon - \sum_{e \in P} (1 - X_\epsilon) \epsilon + c_i, \ \forall i, P \\
X_\epsilon \in [0, 1] \\
Y_{Pi} \in [0, 1]

The SAT score for the graph G is \phi(G) = \max_{P \in G} \phi(P), i.e. the maximum score among all s-t paths.

The QoSCV/QoSCCE problem is to find a minimum set S of edges/vertices such that \phi(G \setminus S) ≤ \rho for a given score threshold \rho. The solution edges/vertices are referred as QoS critical edges/vertices respectively. Notice that QoSCV can be readily converted into QoSCCE through a classic technique [13], we only present solutions for QoSCCE first, and include the conversion in Section VI. In this paper, we refer to the s-t path as path for short.

B. IP formulation

By introducing several variables: X_\epsilon = 1 if edge e is NOT removed in the optimal solution, 0 otherwise; Y_{Pi} = 1 if a s-t path P has w_i(P) ≤ c_i, 0 otherwise; a large constant \epsilon = \max_i \{\sum_{e \in P} w_i(e)\}, we can describe QoSCCE by the following integer program in Fig. 1.

The first constraint demands that the remaining graph satisfies the SAT score threshold. The second constraint demands that w.r.t any constraint, for any path P, if no edge of P is removed (∑_{e \in P} (1 - X_\epsilon) = 0) and P satisfies the i-th constraint (Y_{Pi} = 1), then we require w_i(P) ≤ c_i. Otherwise, w_i(P) is unrestricted. The third constraint demands that if no edge of P is removed, and P does not satisfy constraint i, then w_i(P) > c_i.

Solving this integer program can provide the optimal solution to QoSCCE. However, the problem is theoretically intractable based on the following arguments.

C. Hardness

QoSCCE problem is quite challenging and its complexity class is still an open issue. We find that it does not belong to NP class through the following discussions.

Given an edge subset S as a certificate to QoSCCE problem on graph G, the process of verifying this certificate equals to another decision problem QoS-SP on the remaining graph G \setminus S: does there exist a path P ∈ G \setminus S satisfying \phi(P) ≤ \rho? This problem is NP-Complete.

Lemma 2.1: QoS-SP is NP-Complete.

Proof: This proof is straightforwardly by reduction from MCP[2] problem by letting ρ = ∑_{i=1}^{m} λ_i. Then if there exists a solution for the QoS-SP problem, then the path is a feasible path to MCP, otherwise, MCP has no solution. Since MCP is NP-hard, the proof completes.
Therefore, the certificate of QoSC is not verifiable in polynomial time, thus not in the class of NP. So it is quite challenging to be tackled.

III. EXACT SOLUTION MFMCSP

In real application scenarios, the set of constraints that are taken into account is very limited, which may only consist jitter, packet loss, delay and some others. Therefore, it is quite practical to consider $m$ as a not quite large value, which gives rise to an exact solution called MFMCSP.

The basic idea is to first enumerate all the possible satisfiable combinations of constraints, for example $(c_1, c_3, c_5)$ if $\lambda_1 + \lambda_3 + \lambda_5 \geq \rho$. Therefore $s-t$ paths that satisfy all the constraints within such a combination is a satisfiable path. We only consider the set of minimal constraint combinations, since the set of paths satisfying a set of constraints is surely the superset of those satisfying a superset of these constraints. (set of paths satisfying $(c_1, c_3)$ of course contains those satisfying $(c_1, c_3, c_5)$.) With this set of minimal combinations, we revise the classic Edmonds – Karp algorithm [7] to find the minimum size of edge cut to cut all the augmenting $s-t$ paths which at the same time satisfy any of these combinations. The pseudocode of this algorithm is included in Algorithm 1 where we employ the A* MCSP algorithm in [6] to discover the shortest path that satisfies a specific set of constraints, i.e., a combination of constraints enumerated above. The correctness of this exact solution is shown in Theorem 3.1.

Algorithm 1 MFMCSP

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input: directed graph $G = (V, E)$, constraint set $M = {c_1, \ldots, c_m}$, credit vector $(\lambda_1, \lambda_2, \ldots, \lambda_m)$, satisfiable score threshold $\rho$;</td>
</tr>
<tr>
<td>2</td>
<td>Output: solution set of edge of QoSC;</td>
</tr>
<tr>
<td>3</td>
<td>$S$ ← all the minimal combinations $ss$ of $M$ with $\sum_{i \in ss} \lambda_i &gt; \rho$;</td>
</tr>
<tr>
<td>4</td>
<td>for each edge $(i, j) \in E$ do</td>
</tr>
<tr>
<td>5</td>
<td>$f(i, j) := f(j, i) = 0$;</td>
</tr>
<tr>
<td>6</td>
<td>$c_f(i, j) = 1$ and $c_f(j, i) = 0$.</td>
</tr>
<tr>
<td>7</td>
<td>end for</td>
</tr>
<tr>
<td>8</td>
<td>while $S \neq \emptyset$ do</td>
</tr>
<tr>
<td>9</td>
<td>$ss$ ← extracted from $S$;</td>
</tr>
<tr>
<td>10</td>
<td>while $3q$ ← the shortest path satisfying all the constraints in $ss$ do</td>
</tr>
<tr>
<td>11</td>
<td>for each edge $(u, v) \in q$ do</td>
</tr>
<tr>
<td>12</td>
<td>$c_f(q) := \min(c_f(u, v) : (u, v) \in q)$;</td>
</tr>
<tr>
<td>13</td>
<td>$f(u, v) = f(u, v) + c_f(q)$; $f(v, u) = -f(u, v)$;</td>
</tr>
<tr>
<td>14</td>
<td>$c_f(u, v) = c(u, v) - f(u, v); c_f(v, u) = c(v, u) - f(v, u)$;</td>
</tr>
<tr>
<td>15</td>
<td>end for</td>
</tr>
<tr>
<td>16</td>
<td>end while</td>
</tr>
<tr>
<td>17</td>
<td>end while</td>
</tr>
<tr>
<td>18</td>
<td>all the vertices reachable from $s$ on the residual network induces a cut $T$.</td>
</tr>
<tr>
<td>19</td>
<td>Return $T$.</td>
</tr>
</tbody>
</table>

Theorem 3.1: The edge cut returned by MFMCSP is an optimal solution of QoSC.

Proof: As shown by Fig. 2, the satisfiable paths are within the dotted circle, while others are outside the circle. As stated in MFMCSP, all the satisfiable combinations of constraints are enumerated and based on the argument about the minimal combination, so each of the satisfiable paths should satisfy at least one minimal combination in $S$.

Consider the subgraph induced by all these satisfiable paths as $G'$. It is evident that no satisfiable path will be augmenting after the loop 8-17, otherwise, it is supposed to be discovered in Line 10. Since only the satisfiable paths are augmented within this loop, it can be regarded as a discovery of max-flow on the induced graph $G'$. Therefore, the cut returned $T$ is a min cut of $G'$, denoted as $m(G')$. The equivalence of $m(G')$ to $opt(QoSC)$ is proved by contradiction.

Assume there exists an edge $e \in opt(QoSC)$, while $e$ does not belong to any satisfiable paths, as shown in Fig. 2. Then adding $e$ back to $G$ will not bring back any satisfiable paths, while decreases the optimal solution and draws the contradiction. Therefore, $opt(QoSC)$ is a subset of $G'$. Moreover since removing all edges in $opt(QoSC)$ will disconnect all satisfiable paths, i.e. all $s-t$ path in $G'$, $opt(QoSC)$ is an edge cut of $G'$.

On the other hand, any cut of $G'$ is a feasible solution to QoSC. Suppose that the min cut of $G'$ is smaller than $opt(QoSC)$, then this min cut becomes a smaller optimal solution to QoSC, which draws a contradiction. Otherwise, if the min cut is larger than $opt(QoSC)$, then $opt(QoSC)$ becomes a new min cut.

Therefore, the min cut $T$ is an optimal solution to QoSC.

IV. HEURISTIC SOLUTION SDOP

To handle general network scenarios with arbitrary number of QoS constraints, we provide an efficient heuristic algorithm in this section. We first define a relaxed metric $\varphi(e)$ for each edge $e$, and propose an algorithm called SAT_TEST (Algorithm 2) to approximately decide if there exist a satisfactory path in the remaining graph.

Algorithm 2 SAT_TEST

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input: directed graph $G = (V, E)$, constant $\rho$;</td>
</tr>
<tr>
<td>2</td>
<td>Output: YES if a satisfactory path probably exists, NO otherwise.</td>
</tr>
<tr>
<td>3</td>
<td>for every edge $e \in E$ do</td>
</tr>
<tr>
<td>4</td>
<td>$\varphi(e) := \sum_{i=1}^{m} u_i \lambda_i$;</td>
</tr>
<tr>
<td>5</td>
<td>end for</td>
</tr>
<tr>
<td>6</td>
<td>$p$ ← shortest $s-t$ path on metric $\varphi$;</td>
</tr>
<tr>
<td>7</td>
<td>if $\varphi(p) &gt; \rho$ then</td>
</tr>
<tr>
<td>8</td>
<td>Return NO;</td>
</tr>
<tr>
<td>9</td>
<td>else</td>
</tr>
<tr>
<td>10</td>
<td>Return YES;</td>
</tr>
<tr>
<td>11</td>
<td>end if</td>
</tr>
</tbody>
</table>

Based on the relaxed test, considering that the shortest path w.r.t each measure (single metric shortest path) is more likely to be a satisfiable path, we count a betweenness metric, i.e., number of appearances of each edge in such path kind, and remove the edge with the greatest number of appearances one by one until the relaxed test returns NO. The greedy heuristic is called SDOP and included in Algorithm 3.
Algorithm 3 SDOP

1: Input: directed graph $G = (V, E)$, constant $\rho$;
2: Output: a set $D$ of edges to be removed.
3: Set $D \leftarrow \emptyset$;
4: while SAT_TEST($G$) == YES do
5:   find all $m$ single metric shortest paths $\{p_1, \cdots, p_m\}$
6:   for all edges $e \in E$ do
7:      Find the one appears in the maximum number of such path;
8:    end for
9:    $D \leftarrow D \cup \{e\}$; $E \leftarrow E \setminus \{e\}$;
10: end while
11: Return $D$.

V. PERFORMANCE EVALUATIONS

Despite the NP-hardness shown in the section II, through extensive simulations on multiple real and synthetic network topologies in this section, our provided solutions are both effective and efficient in terms of evaluation accuracy and time complexity. With regard to the topologies adopted, we use a well-known Internet topology generator BRITE [8] to generate topologies at three different internet levels: Flat Router-Level only, Flat AS-Level only, Hierarchical Top-down that follow two models: Power-law and Waxman. In principle, the simulations are of three-fold: (1) to show the efficiency of vulnerability assessment of Algorithm 1 for network with moderate constraint amount, we study the number of QoS-critical edges w.r.t various threshold levels of SAT scores (which are referred as SAT rate defined later), by executing MFMCSP on networks following the random models mentioned above, where the time complexity results are also included; (2) to show the accuracy of the proposed heuristic SDOP for vulnerability assessment, we conduct the algorithm on the same four networks as we used MFMCSP, to compare their assessment result and time complexity; (3) to show the scalability of SDOP and the proposed framework, we run on a generate power-law network with increasing size as well as constraint amount and report the corresponding time costs.

Specifically, as in [10][11], all edge weights are generated following Uniform Distribution within the range $[1, 10]$ and the each dimension of the constraint priority vector $(\lambda_1, \cdots, \lambda_m)$ obeys the same distribution. We call $\sum_{i=1}^{m} \lambda_i$ as the full score of the given topology, and in order to guarantee that the QoS-optimal path of the topology can achieve such a full score before any edges are removed, we use the weight vector of a shortest path w.r.t hop count in the given topology as the constraint vector. The satisfactory rate is defined as the ratio of the threshold over full score, i.e. $\rho / \sum_{i=1}^{m} \lambda_i$. All the simulation results are averaged from 30 instances. All the tests are implemented by C++ and performed on a 2.33GHz Linux Workstation with 8GB RAM.

Fig. 3 reports the solution sizes of QoSCE detected by MFMCSP on six 500-node 1500-edge networks with moderate $m = 5$, as shown. For each model, the solution size of QoSCE w.r.t satisfactory rates ranging with $[0.05, 0.95]$ with a step 0.05 are presented. Since larger QoS solution sizes indicates a higher vulnerability, we can derive the following rough sequence with the QoS vulnerability, which starts with the most robust topology: Router-PowerLaw > AS-Waxman > Hierarchical- Waxman > Router- Waxman > Hierarchical-Powerlaw > AS-Powerlaw. Based on this evaluation, for Router-only level QoS-sensitive networks, PowerLaw model is more robust, while for AS-only level networks, Waxman model is a better choice. Another observation is that the QoSC solution sizes of most network topologies are quite stable despite the changes of satisfactory rate. This dues to the uniformly distributed edge weights assumed, where a large amount of $s - t$ paths exist and have similar satisfactory scores, therefore, all these paths are required to be cut and the minimum $s - t$ cut of the graph becomes the optimal solution of QoSCE, which remains unchanged as the satisfactory rate decreases. The time cost of MFMCSP on these topologies are bounded by around 15 seconds, as shown in Fig.4. As the satisfactory rate decreases, the time complexity increases because more satisfactory paths are discovered and flow-augmented. Though this network has only 1500 edges, since the flow-augmenting process in MFMCSP only visits each edge once, this exact solution can also be applied to a large-scale network. However, the proposed heuristic SDOP is shown to achieve the similar performance vulnerability assessment with much less time expense, so we do not test MFMCSP on large network.

Fig. 5 presents the QoSCE solution sizes derived by SDOP for the same six topologies as above. Although the solution sizes are larger than the optimal values returned by MFMCSP, it provides the same robustness sequence. Therefore, the SAT_TEST does provide a good estimation over the network. Furthermore, the time expense of the assessment is reduced from 15 seconds to 2 seconds by use of this algorithm.
VI. FURTHER DISCUSSIONS AND CONCLUSIONS

For small-size networks, an exact solution can be proposed by leveraging the concept of Pareto Optimal (PO) Path for solving shortest path problems [4][2][5][3]: A path $p$ is Pareto Optimal iff $p$ is not dominated by any other paths, i.e. there does not exist a path $q$ with $w_i(q) < w_i(p)$ for all $i \in [1, \ldots, m]$. It is evident that in the remaining graph, no satisfiable path exists if $\phi(p) < \rho$ for all PO path $p$. Therefore, instead of enumerating all the satisfiable constraint combination as MFMCS does, we can employ an existing label-correcting algorithm [4] to find the the PO path $p$ with the maximum $\phi(p)$, which can be plugged into the MFMCS algorithm to terminate the flow-augmenting process and return an exact solution. Although the searching space is compacted by eliminating all non-PO paths, this algorithm still requires further acceleration to apply for large-scale networks. Considering the fruitful researches regarding PO paths, this will become a potential direction for our further work.

As to the QoSCV problem, we can readily covert it into an instance of QoSCE: for each vertex $u \in V$, replace $u$ with two new vertices $u^1$ and $u^2$, then add an edges $(v, u^1)$ for every edge $(v, u) \in E$ and an edge $(u^2, w)$ for every edge $(u, w) \in E$. Set $w(v, u^1) = w(v, u)$, $w(u^2, w) = w(u, w)$, $w(u^1, u^2) = 0$, $c_f(u^1, u^2) = 1$ and $c_f(u^2, u^1) = 0$. Then executing MFMCS algorithm on this instance returns an exact solution for QoSCV.

In summary, we propose the first QoS-aware assessment framework for network topology vulnerabilities, which is formulated as a graph optimization problem whose theoretical intractability is shown. By providing one exact solution and one efficient heuristic, the scalability of this framework is validated through extensive simulation studies over several popular network models. This work also provides an insight of other possible solutions, which benefits further researches.

Acknowledgement

This work is partially supported by NSF Career Award # 0953284 and DTRA, Young Investigator Award, Basic Research Program # HDTRA1-09-1-0061.

REFERENCES
