Robust validation of network designs under uncertain demands and failures

Yiyang Chang, Sanjay Rao, and Mohit Tawarmalani
Purdue University

USENIX NSDI 2017
Validating network design

• Network design today is ad-hoc, and validating design is usually an afterthought
  • Contrast: Tools for chip and software industry a $10B business [Mckeown, 2012]

• Much progress on verification of network data plane (e.g., reachability, security policy)
  • HSA, Veriflow, Batfish, NoD, etc.

• Our goal: Validating quantitative network properties
  • Formal approach to guarantee network performance (e.g., bandwidth, link utilization)
    • Under diverse failure/traffic scenarios
  • Use the formal approach to inform network design
Why is network validation hard? (1)

• Scenarios of interest are too many
  • Exponentially many failure scenarios [Wang et al., Sigcomm ’10, Liu et al., Sigcomm ’14]
    • E.g., All possible simultaneous $f$ link failures
  • All possible traffic demands — non-enumerable
Why is network validation hard? (2)

- Adaptation makes the problem intractable
- Networks increasingly agile and flexible in adaptation
  - E.g., SDNs and NFVs
- Tools exist to bound worst case performance
  - E.g., robust optimization, and oblivious routing
    [Applegate et al., Sigcomm ’03]
  - Assume networks do not adapt, or consider limited forms of adaptation to make problem tractable
Our work

- General framework for network validation
  - Find the worst performance of the network across all scenarios assuming network can adapt in best fashion for each scenario

- Handles intractable problems drawing on cutting-edge optimization technique

- Applies to network synthesis

Worst performance = max\{m_1, m_2, \ldots, m_N\}
Less is better
### Example: Failure validation

<table>
<thead>
<tr>
<th>Uncertainty Set</th>
<th>Adaptations</th>
<th>Performance metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>• All ( f ) or fewer link failures</td>
<td>• Flexible rerouting (multi-commodity flow)</td>
<td>• Utilization of most congested link</td>
</tr>
</tbody>
</table>

**Problem:**

• Given up to \( f \) links may simultaneously fail, what is the worst-case utilization of any link across all failure scenarios?
Formal formulation of a network validation problem

\[
\max_{x \in X} \quad \min_{y \in Y(x)} \quad F(x, y)
\]

**Uncertainty Set**

**Adaptations**

**Performance metric**

Less is better

---

**Example: Validation under failures**

<table>
<thead>
<tr>
<th>$X$</th>
<th>Set of failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y(x)$</td>
<td>Feasible routing of demands under given failure</td>
</tr>
<tr>
<td>$F(x, y)$</td>
<td>Utilization of most congested link</td>
</tr>
</tbody>
</table>

**Inner problem:** For a fixed scenario - Easy to compute online (LP)

E.g., multi-commodity flow

**Outer problem:** Potentially hard since large number of scenarios
## Wide applicability of framework

<table>
<thead>
<tr>
<th>Uncertainty Set</th>
<th>Adaptations</th>
<th>Performance metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>• All $f$ or fewer link failures</td>
<td>• Flexible rerouting (multi-commodity flow)</td>
<td>• Utilization of most congested link</td>
</tr>
<tr>
<td>• Shared risk link group</td>
<td>• Rerouting constrained to pre-selected tunnels</td>
<td>• Bandwidth of business critical applications</td>
</tr>
<tr>
<td>• Weighted averages of historical demands</td>
<td>• Constrain with middlebox traversal requirements</td>
<td></td>
</tr>
</tbody>
</table>
Reformulating the problem

\[ \max_{x \in X} \min_{y \in Y(x)} F(x, y) \]

LP dualization

\[ \max_{\lambda, v, x} F'(\lambda, v, x) \]
Failure validation: Formulation

\[
\begin{align*}
\max_{v, \lambda, x} & \quad \sum_{t, i \neq t} d_{it} (v_{it} - v_{tt}) \\
\text{s.t.} & \quad v_{it} - v_{jt} \leq \lambda_{ij} \quad \forall t, \langle i, j \rangle \in E \\
& \quad \sum_{\langle i, j \rangle \in E} \lambda_{ij} c_{ij} (1 - x_{ij}^f) = 1 \\
& \quad x^f \in X \\
& \quad x_{ij}^f \in \{0, 1\}; \quad \lambda_{ij} \geq 0; \quad \langle i, j \rangle \in E
\end{align*}
\]
Depends on failure model of interests
• E.g. simultaneous $f$ link failures
Failure validation: Formulation

\[
\begin{align*}
    \max_{v,\lambda,x} \quad & \sum_{t,i \neq t} d_{it} (v_{it} - v_{tt}) \\
    \text{s.t.} \quad & v_{it} - v_{jt} \leq \lambda_{ij} \quad \forall t, \langle i, j \rangle \in E \\
        & \sum_{\langle i, j \rangle \in E} \lambda_{ij} c_{ij} (1 - x^f_{ij}) = 1 \\
        & x^f \in X \\
        & x^f_{ij} \in \{0, 1\}; \quad \lambda_{ij} \geq 0; \quad \langle i, j \rangle \in E
\end{align*}
\]

Can be converted to mixed-inter linear program. In general, validation problems could be non-linear.
Solution approach

- Focus on upper bounds (relaxation)
  - Intractable problems – hard to solve to optimality
  - Upper bounds sufficient for validation use

- Goal: Develop a general approach
  - Applicable to diverse validation problems (e.g., validating failures, demands...)
  - Yet, amenable to problem-specific structure

- Use cutting-edge techniques from non-linear optimization
Tractable relaxations: RLT

- **RLT relaxations**: general approach to relax non-convex problems into tractable LPs
  - Family of relaxations
  - Higher levels of hierarchy
    - Converge to optimal value of the non-convex problem
    - Incur higher complexity
  - For scalability, focus on the first level
RLT relaxation: example

\[
\begin{align*}
\min_{x,y} & \quad [xy] - x + y \\
& \quad [x - 2 \geq 0; \quad y - 3 \geq 0] \\
& \quad 3 - x \geq 0; \quad 4 - y \geq 0
\end{align*}
\]

\[ \rightarrow \quad z = x + y \]

\[ \rightarrow \quad xy - 2y - 3x + 6 \geq 0 \]

Relaxation steps:
1. Multiply constraints with each other
2. Replace products of variables xy, x^2, y^2 by new variables
Our results on effectiveness of RLT

- Compare RLT with two theoretical benchmarks
  - Both bound worst case performance across failures/demands, but with limited network adaptation
  - Oblivious routing [Applegate, et al., Sigcomm ’03; Wang, et al., Sigcomm ’06, etc.]
  - Affine adaptation: a generalization of oblivious routing, studied in robust optimization

- Our results show
  - First-level RLT dominate oblivious/affine adaptations
  - Better results possible by exploiting problem-specific structure combined with RLT
Evaluation

• Real topologies
  • Abilene, GEANT, and ANS (from The Internet Topology Zoo)

• Real and synthetic traffic matrices
  • Real trace: 6-month end-to-end demand on Abilene
  • Synthetic: Gravity model
Results: Effectiveness of RLT

- Compare maximum link utilization (MLU)
- The optimal IP scheme vs. our RLT relaxation LP
- RLT matches optimal in all our experiments

Abilene Network — 3 link failures
Results: Effectiveness of RLT

- Compare with R3 [Wang et al., Sigcomm ’10]
  - Determines if MLU < 1 under f failures
  - Gives a valid bound only when MLU < 1
  - Based on oblivious approach

- Our result
  - First-level RLT dominates R3 whenever R3 provides a valid bound

- Other advantages of our approach
  - Useful to detect bad failure scenarios, and the amount of exceeded link capacity
  - Generalizes to other validation problems

Abilene Network — 3 link failures
Using framework to detect bad failures

- Framework allows finding failures that impact the network the most
- Random search not efficient
- Only 0.05% of 3-failure scenarios are bad (MLU > 1)
- Emulate to understand latency behavior

Emulated Abilene traffic matrix with Mininet, and ONOS controller
Results: running time

• RLT relaxation LP vs. optimal IP (IP run for 2 hours)

• On scaled GEANT network (32 nodes, 1000 edges), 3 link failures:
  • RLT finished in 608 seconds, whereas IP finished in 3890 seconds
  • Only 60% of the IP instances completed in 2 hours

• Our RLT relaxation LP doesn’t degrade with larger number of failures
Example: Tunnel selection validation

Uncertainty Set
- All \( f \) or fewer link failures
- Shared risk link group
- Weighted averages of historical demands

Adaptations
- Flexibly rerouting (Multi-commodity flow)
- Rerouting constrained to pre-selected tunnels
- Constrain with middlebox traversal requirements

Performance metric
- Utilization of most congested link
- Bandwidth of business critical applications

Problem:
- For a given choice of tunnels, are utilizations of all links across all traffic demands of interest within acceptable limits?
Tunnel selection: Results

• **Predicted demand**: weighted averages of historical matrices
  • Validation problem is an LP
  • On Abilene: First-level RLT achieves optimal MLU

• Widely-used tunnel selection heuristics may perform poorly
  • E.g., K-shortest (SWAN, Sigcomm ’13), Shortest-Disjoint heuristics
  • More robust tunnel selection heuristic performs much better
Synthesizing valid designs

• Validation is a stepping stone for synthesis

• Example: Optimal Capacity Augmentation
  • Incrementally add capacity to existing links
  • Minimizing cost of adding capacity
  • Ensure resulting network can handle all failure scenarios

• One can use our framework for synthesis in 2 ways:
  1) Get conservative solution, with a single LP
  2) Iterative approach, which gives a lower bound on cost at each step
Capacity augmentation: Abilene

- Validate if MLU $\leq 1$.
- If not, run augmentation LP with counter examples
Capacity augmentation: Abilene

- Validate if $\text{MLU} \leq 1$.
- If not, run augmentation LP with counter examples.

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Capacity augmentation: Abilene

- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td>(1, 10)</td>
<td>2.744</td>
</tr>
</tbody>
</table>
Capacity augmentation: Abilene

- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td>(1, 10)</td>
<td>2.744</td>
</tr>
<tr>
<td>2</td>
<td>(2, 9), (10, 1)</td>
<td>1.274</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Capacity augmentation: Abilene

- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td>(1, 10)</td>
<td>2.744</td>
</tr>
<tr>
<td>2</td>
<td>(2, 9), (10, 1)</td>
<td>1.274</td>
<td>(2, 9)</td>
<td>5.488</td>
</tr>
</tbody>
</table>
Capacity augmentation: Abilene

- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td>(1, 10)</td>
<td>2.744</td>
</tr>
<tr>
<td>2</td>
<td>(2, 9), (10, 1)</td>
<td>1.274</td>
<td>(2, 9)</td>
<td>5.488</td>
</tr>
<tr>
<td>3</td>
<td>(9, 8), (10, 7)</td>
<td>1.217</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Capacity augmentation: Abilene

- Validate if MLU $\leq 1$.
- If not, run augmentation LP with counter examples

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td>(1, 10)</td>
<td>2.744</td>
</tr>
<tr>
<td>2</td>
<td>(2, 9), (10, 1)</td>
<td>1.274</td>
<td>(2, 9)</td>
<td>5.488</td>
</tr>
<tr>
<td>3</td>
<td>(9, 8), (10, 7)</td>
<td>1.217</td>
<td>(9, 8)</td>
<td>7.653</td>
</tr>
</tbody>
</table>
Capacity augmentation: Abilene

- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter examples</th>
<th>MLU</th>
<th>Links Augmented</th>
<th>Total new capacity (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 10), (2, 9)</td>
<td>1.274</td>
<td>(1, 10)</td>
<td>2.744</td>
</tr>
<tr>
<td>2</td>
<td>(2, 9), (1, 10)</td>
<td>1.274</td>
<td>(2, 9)</td>
<td>5.488</td>
</tr>
<tr>
<td>3</td>
<td>(9, 8), (10, 7)</td>
<td>1.217</td>
<td>(9, 8)</td>
<td>7.653</td>
</tr>
<tr>
<td>4</td>
<td>(10, 7), (9, 8)</td>
<td>1.217</td>
<td>(10, 7)</td>
<td>9.818</td>
</tr>
<tr>
<td>5</td>
<td>(0, 2), (1, 10)</td>
<td>1.192</td>
<td>(0, 2)</td>
<td>11.743</td>
</tr>
<tr>
<td>6</td>
<td>(1, 0), (1, 10)</td>
<td>1.071</td>
<td>(1, 0)</td>
<td>12.452</td>
</tr>
<tr>
<td>7</td>
<td>(7, 6), (8, 5)</td>
<td>1.006</td>
<td>(7, 6)</td>
<td>12.509</td>
</tr>
<tr>
<td>8</td>
<td>(8, 5), (7, 6)</td>
<td>1.006</td>
<td>(8, 5)</td>
<td>12.566</td>
</tr>
<tr>
<td>9</td>
<td>—</td>
<td>1.000</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Conclusions

• Early effort at formally verifying quantitative network properties under uncertainty

• Generic framework for a wide class of network validation problems

• Modeling adaptivity results in intractable problems
  • RLT relaxations promising
  • Tighter bounds than oblivious
  • Exact in multiple failures case and predicted demand case

• Validation framework enables network synthesis
Thanks!
Questions?