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)

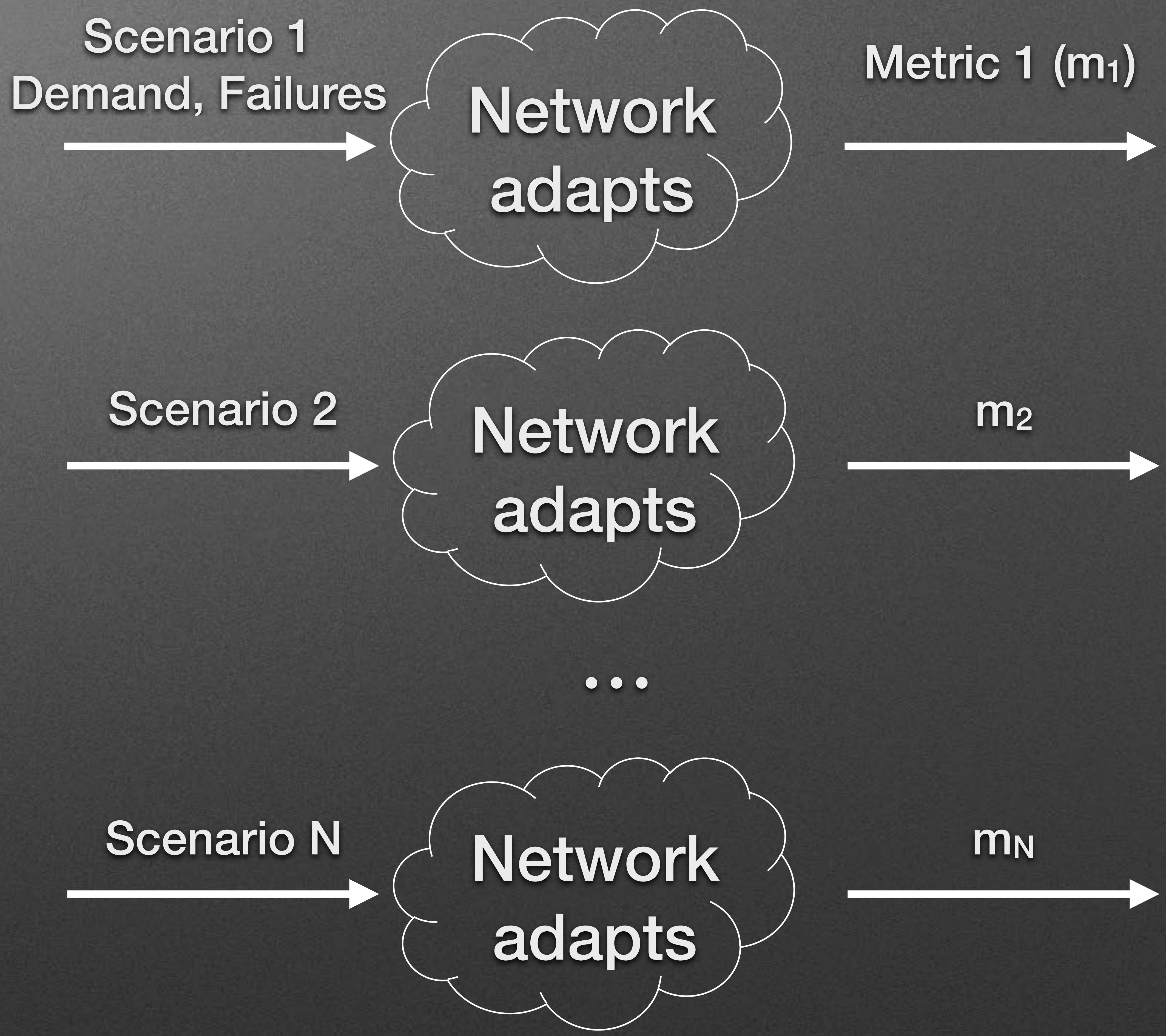
- Scenarios of interest are too many
  - Exponentially many failure scenarios [Wang et al., Sigcomm ’10, Liu et al., Sigcomm ’14]
    - E.g., All possible simultaneous *f* link failures
  - All possible traffic demands — non-enumerable

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

Uncertainty Set

- All *f* or fewer link failures

Adaptations

- Flexible rerouting (multi-commodity flow)

Performance metric

- Utilization of most congested link

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

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

Uncertainty Set — Adaptations — Performance metric (Less is better)

Example: Validation under failures	
X	Set of failures
Y(x)	Feasible routing of demands under given failure
F(x, y)	Utilization of most congested link

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

Uncertainty Set

- All *f* or fewer link failures
- Shared risk link group
- Weighted averages of historical demands

Adaptations

- Flexible 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

Reformulating the problem

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

LP dualization

$$\max_{\lambda, v, x} \quad F'(\lambda, v, x)$$

Failure validation: Formulation

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

Failure validation: Formulation

$$\max_{v,\lambda,x} \quad \sum_{t,i\neq t} d_{it}(v_{it} - v_{tt})$$

$$s.t. \quad v_{it} - v_{jt} \leq \lambda_{ij} \quad \forall t, \langle i,j\rangle \in E$$

$$\sum_{\langle i,j\rangle\in E} \lambda_{ij} c_{ij}(1 - x^f_{ij}) = 1$$

$$x^f \in X$$

$$x^f_{ij} \in \{0,1\}; \quad \lambda_{ij} \geq 0; \quad \langle i,j\rangle \in E$$

Depends on failure model of interests

- E.g. simultaneous *f* link failures

$$\sum_{i,j} x^f_{ij} = f$$

Failure validation: Formulation

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

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

$$\min_{x,y} \quad xy - x + y \quad \rightarrow \quad z - x + y$$

$$x - 2 \geq 0; \quad y - 3 \geq 0 \quad \rightarrow \quad xy - 2y - 3x + 6 \geq 0$$

$$3 - x \geq 0; \quad 4 - y \geq 0$$

$$xy \downarrow z$$

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

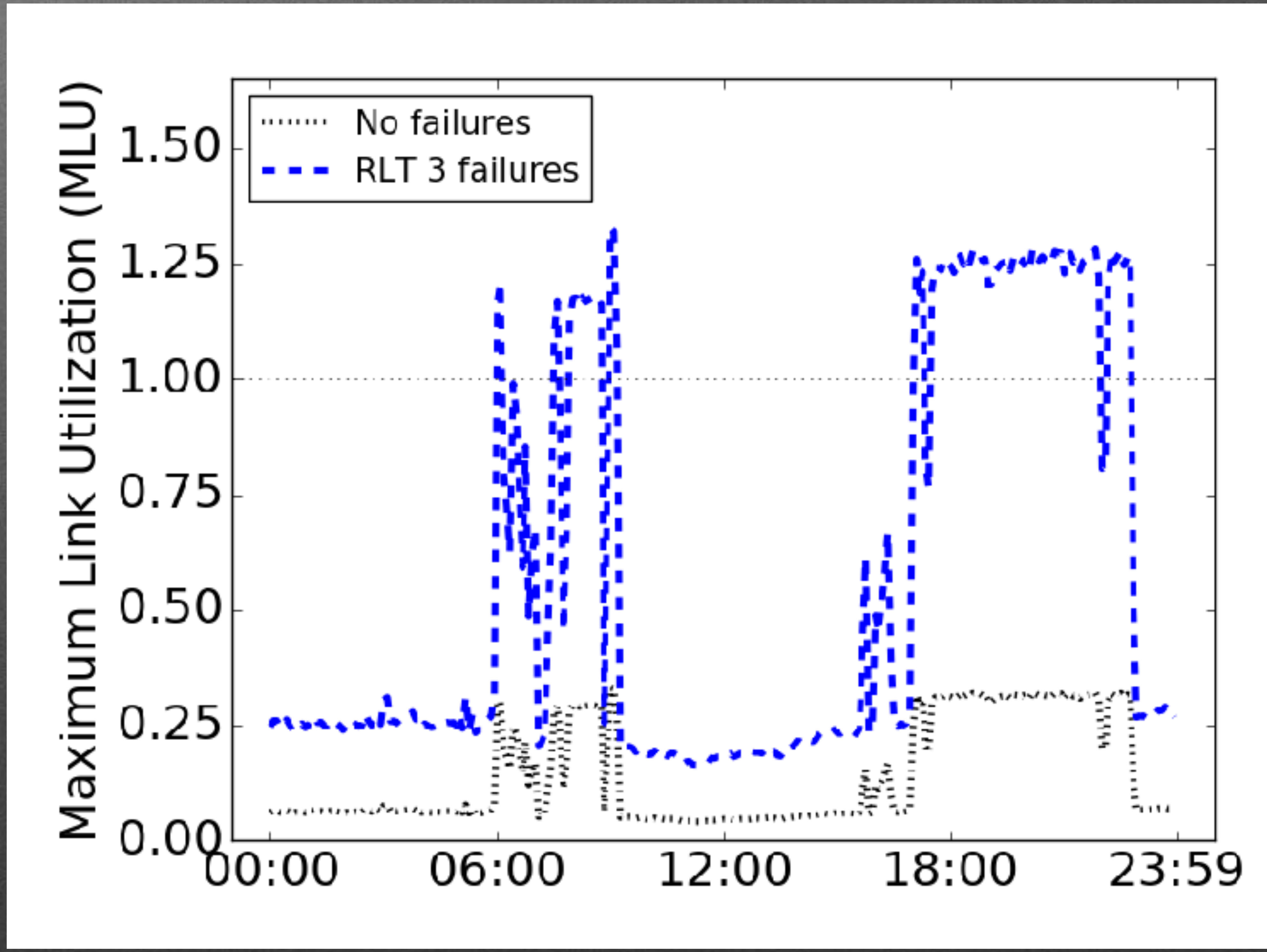
- **Compare RLT with two theoretical benchmarks**
  - Both bound worst case performance across failures/demands, but with limited network adaptation
  - Oblivious routing [Applegate, et al., Sigcomm ’03; Wang, et al., Sigcomm ’06, etc.]
  - Affine adaptation: a generalization of oblivious routing, studied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



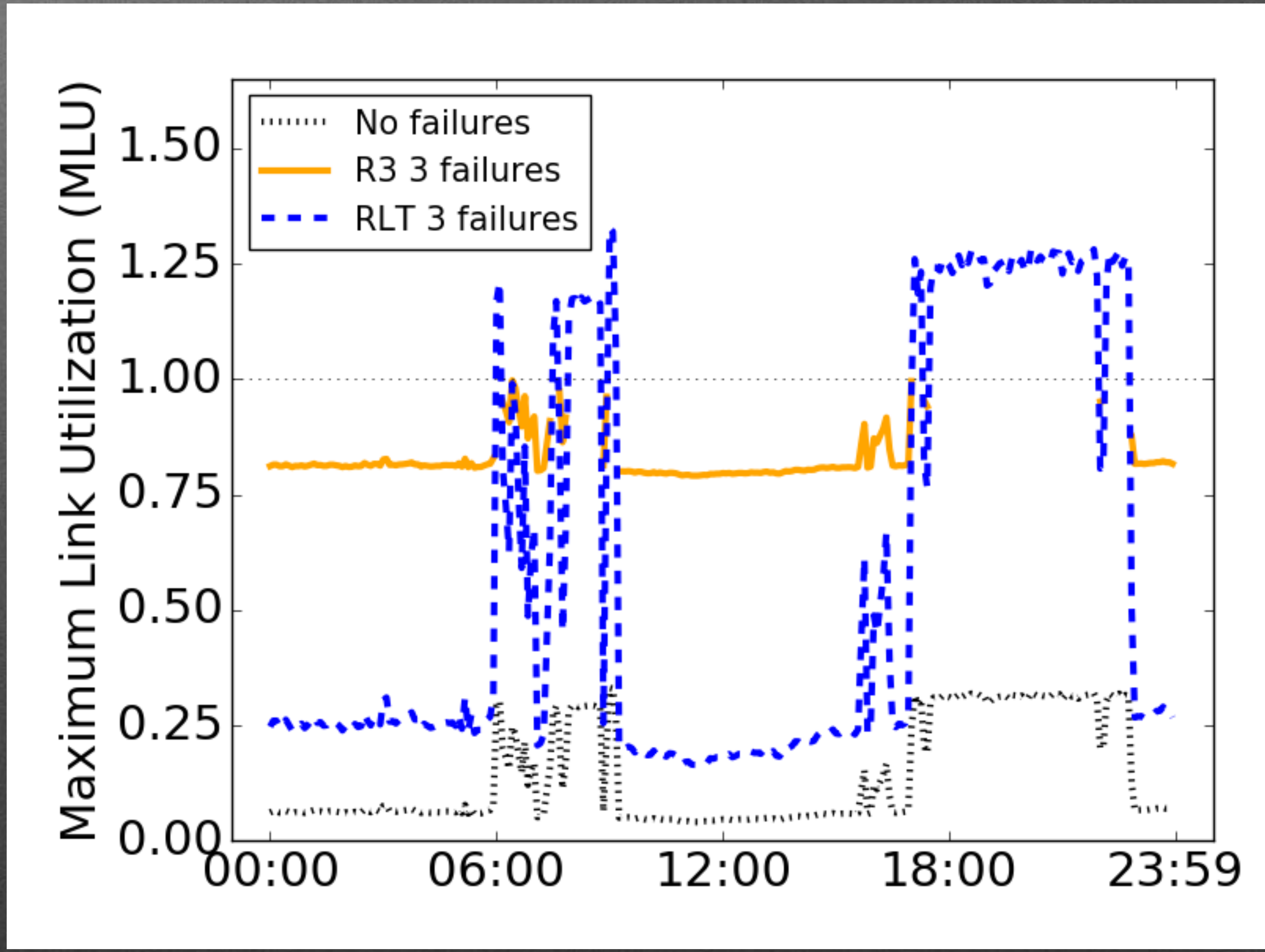
Abilene Network — 3 link failures

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

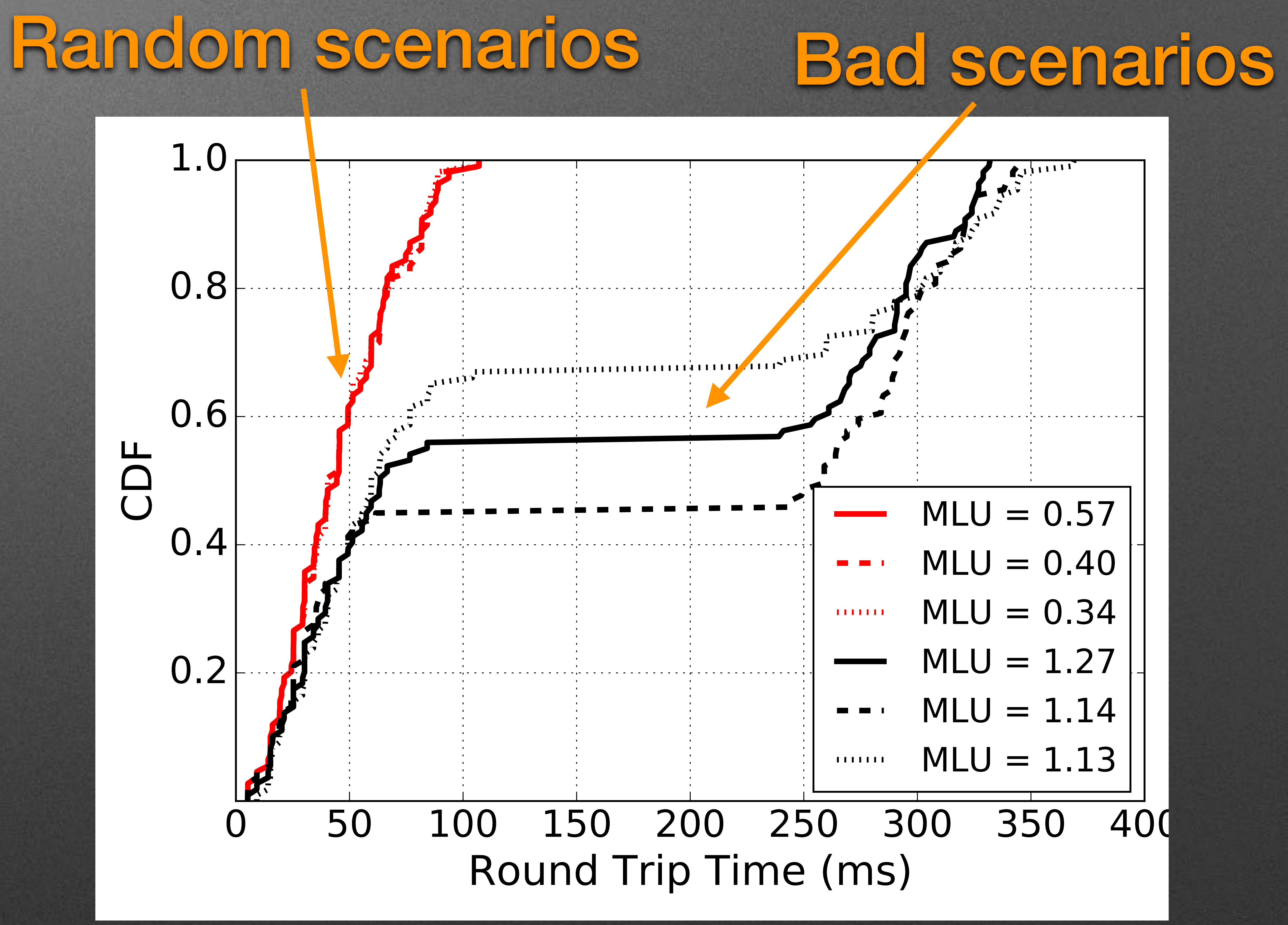


Abilene Network — 3 link failures

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

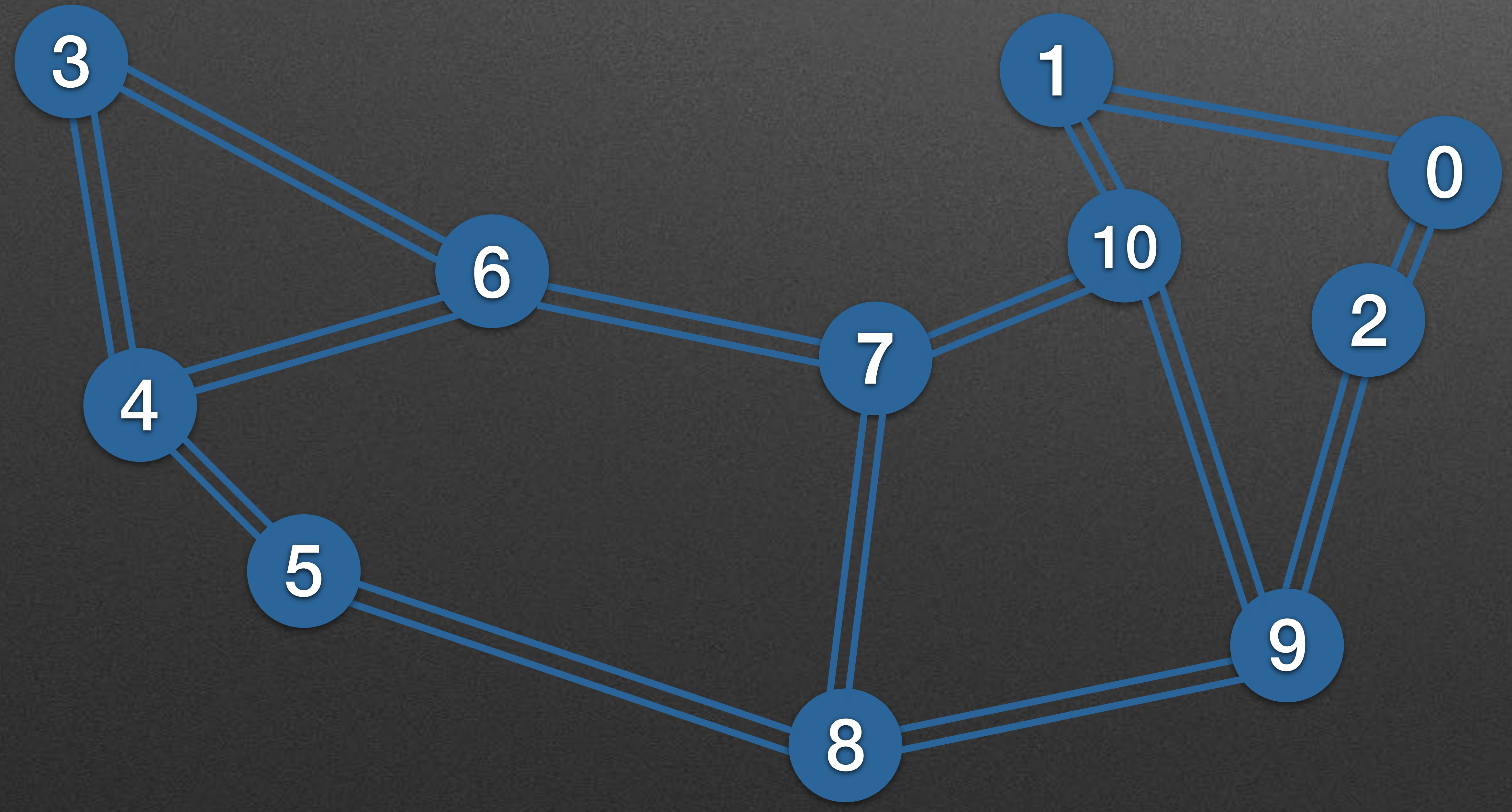
- Predicted demand: weighted averages of historical matrices
  - Validation problem is an LP
  - On Abilene: First-level RLT achieves optimal MLU
- Widely-used tunnel selection heuristics may perform poorly
  - E.g., K-shortest (SWAN, Sigcomm ’13), Shortest-Disjoint heuristics
  - More robust tunnel 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

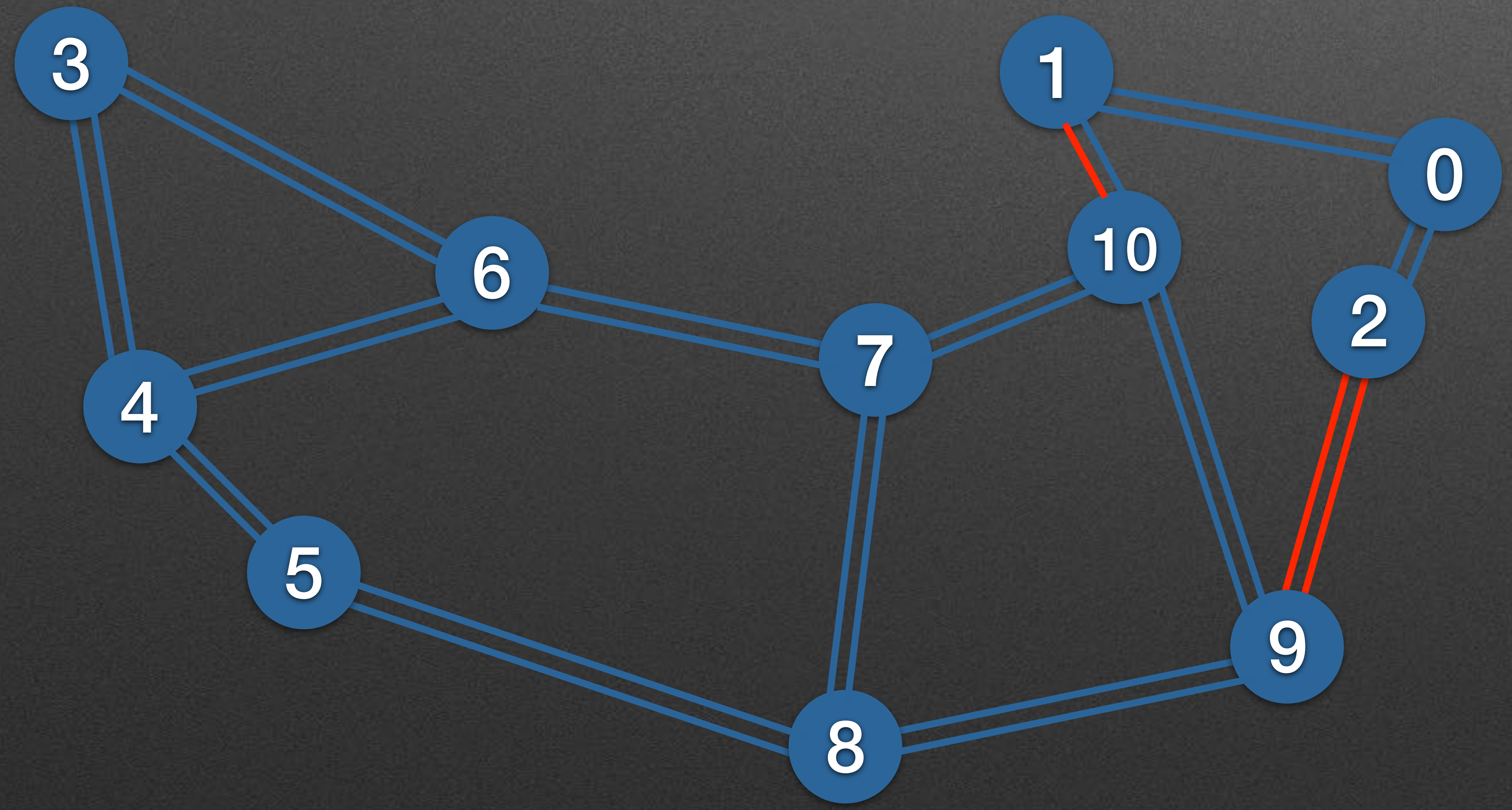
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)

Capacity augmentation: Abilene

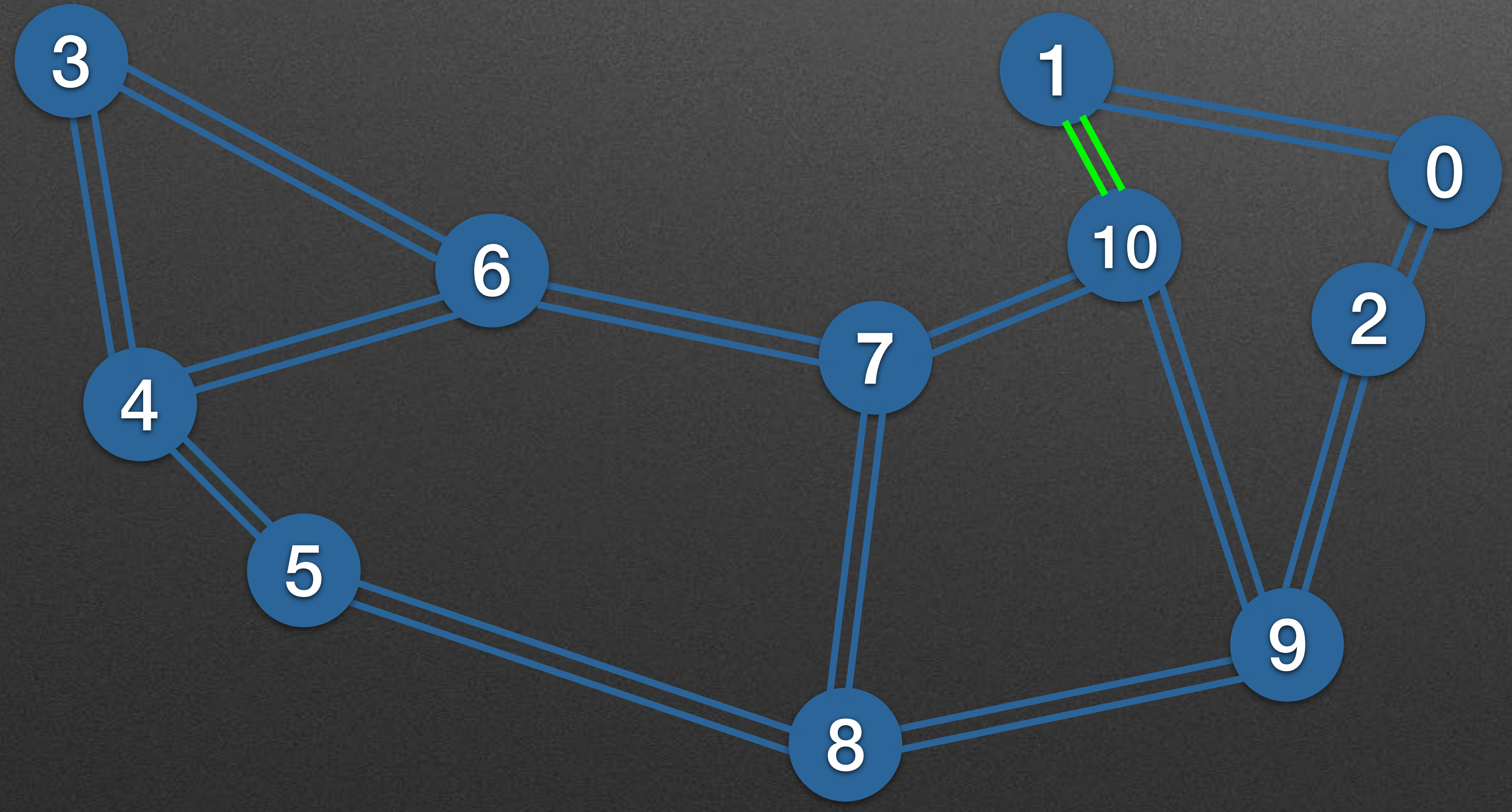
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274		

Capacity augmentation: Abilene

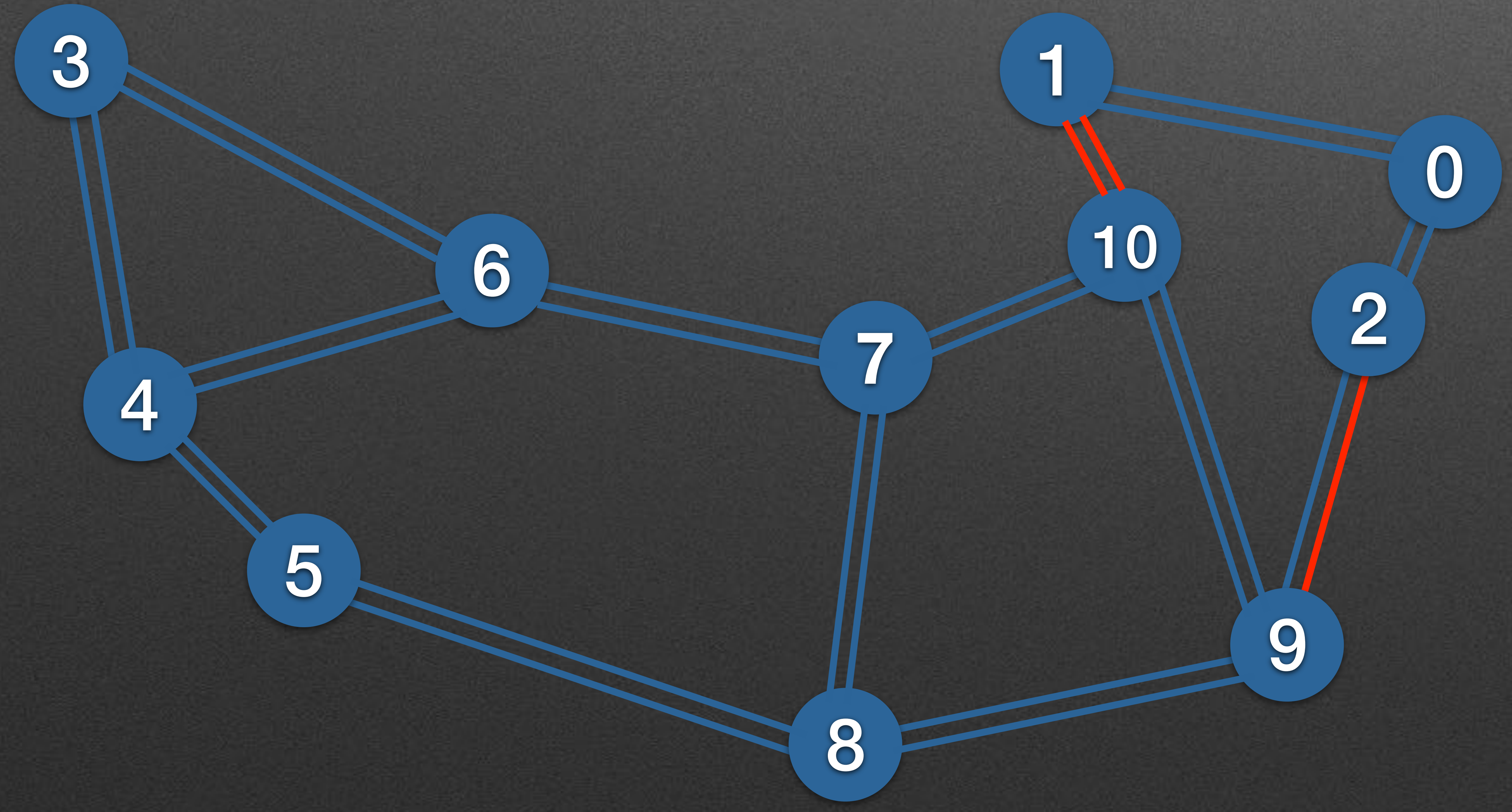
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274	(1, 10)	2.744

Capacity augmentation: Abilene

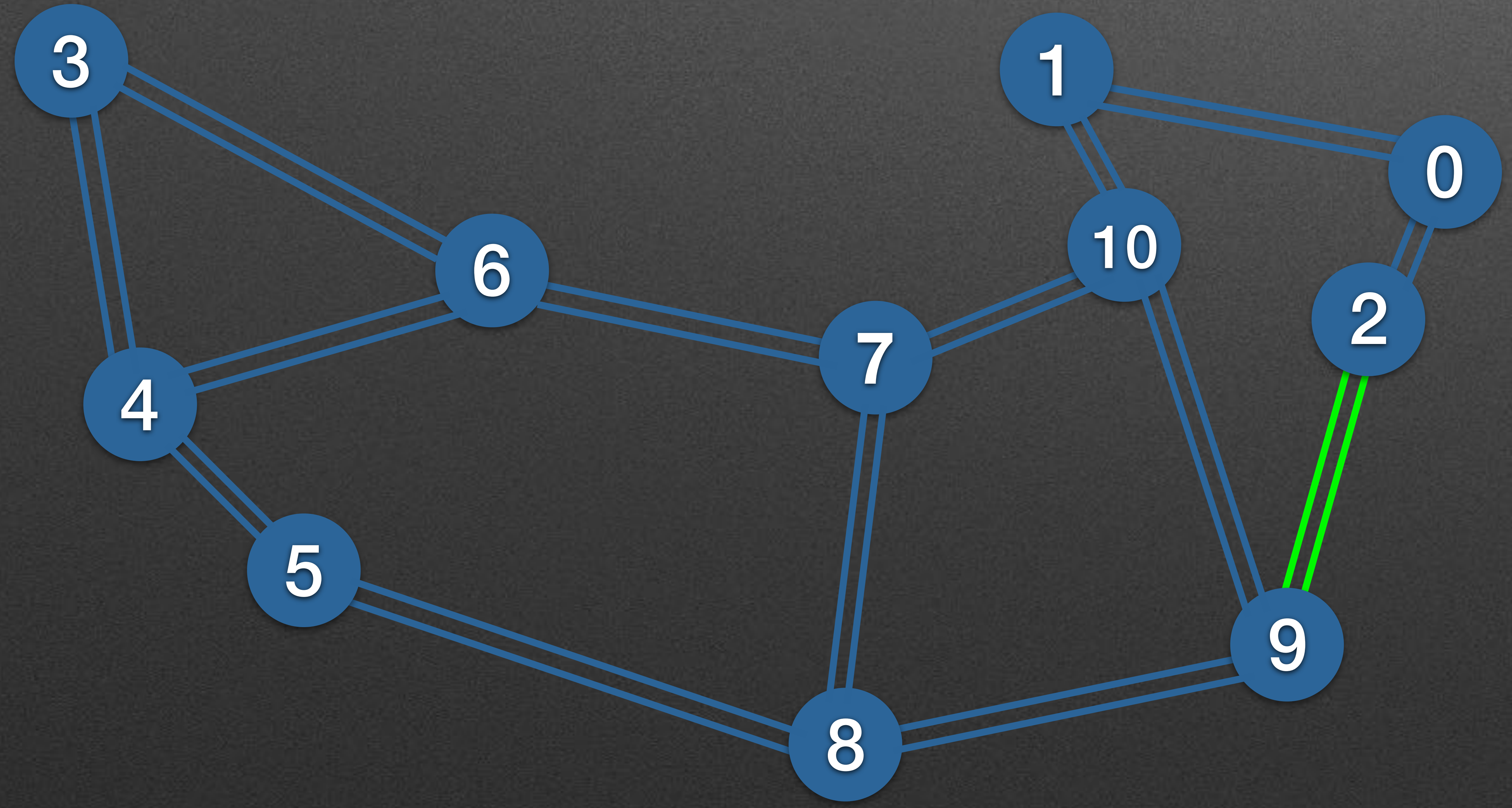
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274	(1, 10)	2.744
2	(2, 9), (10, 1)	1.274		

Capacity augmentation: Abilene

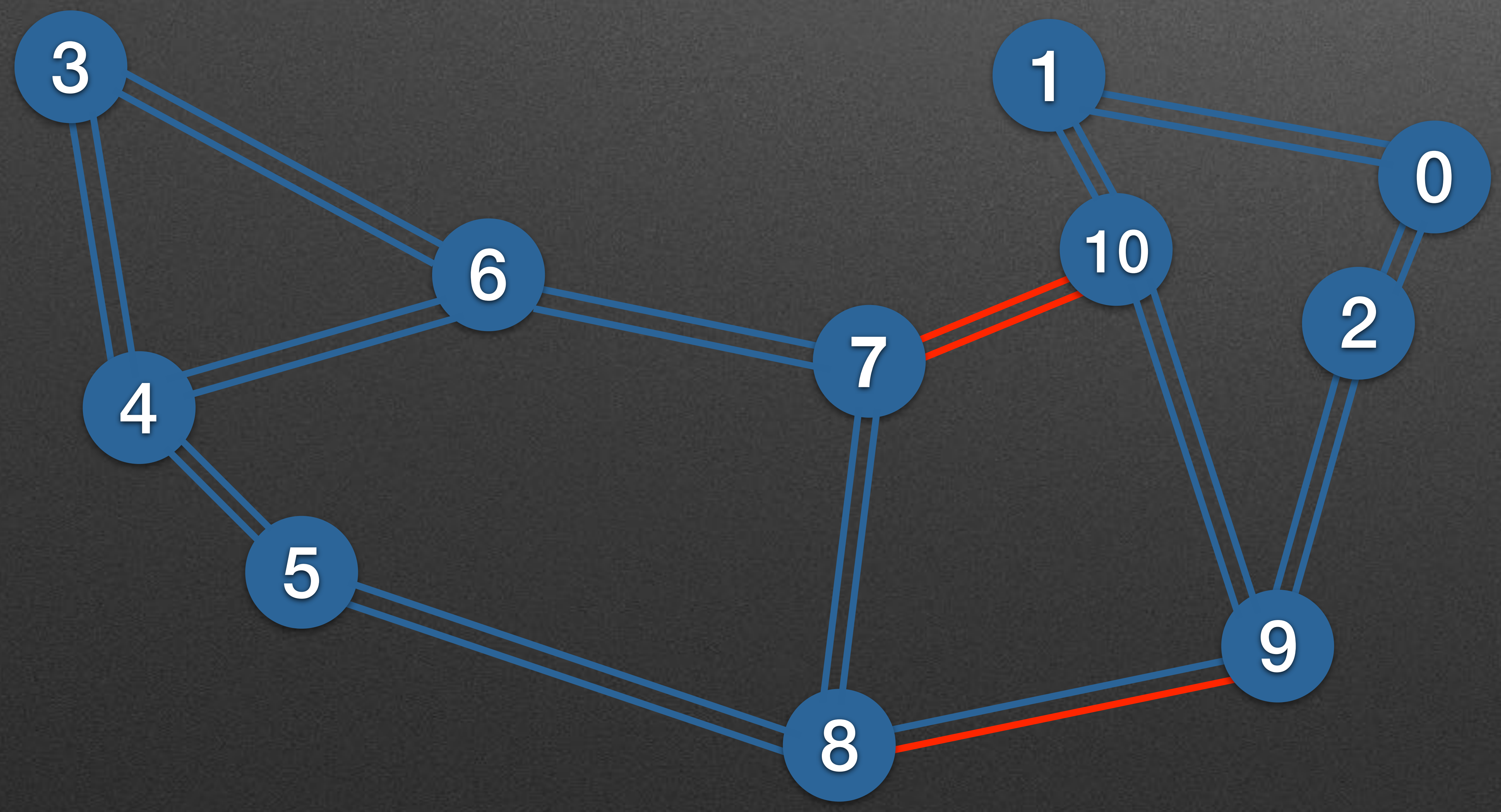
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274	(1, 10)	2.744
2	(2, 9), (10, 1)	1.274	(2, 9)	5.488

Capacity augmentation: Abilene

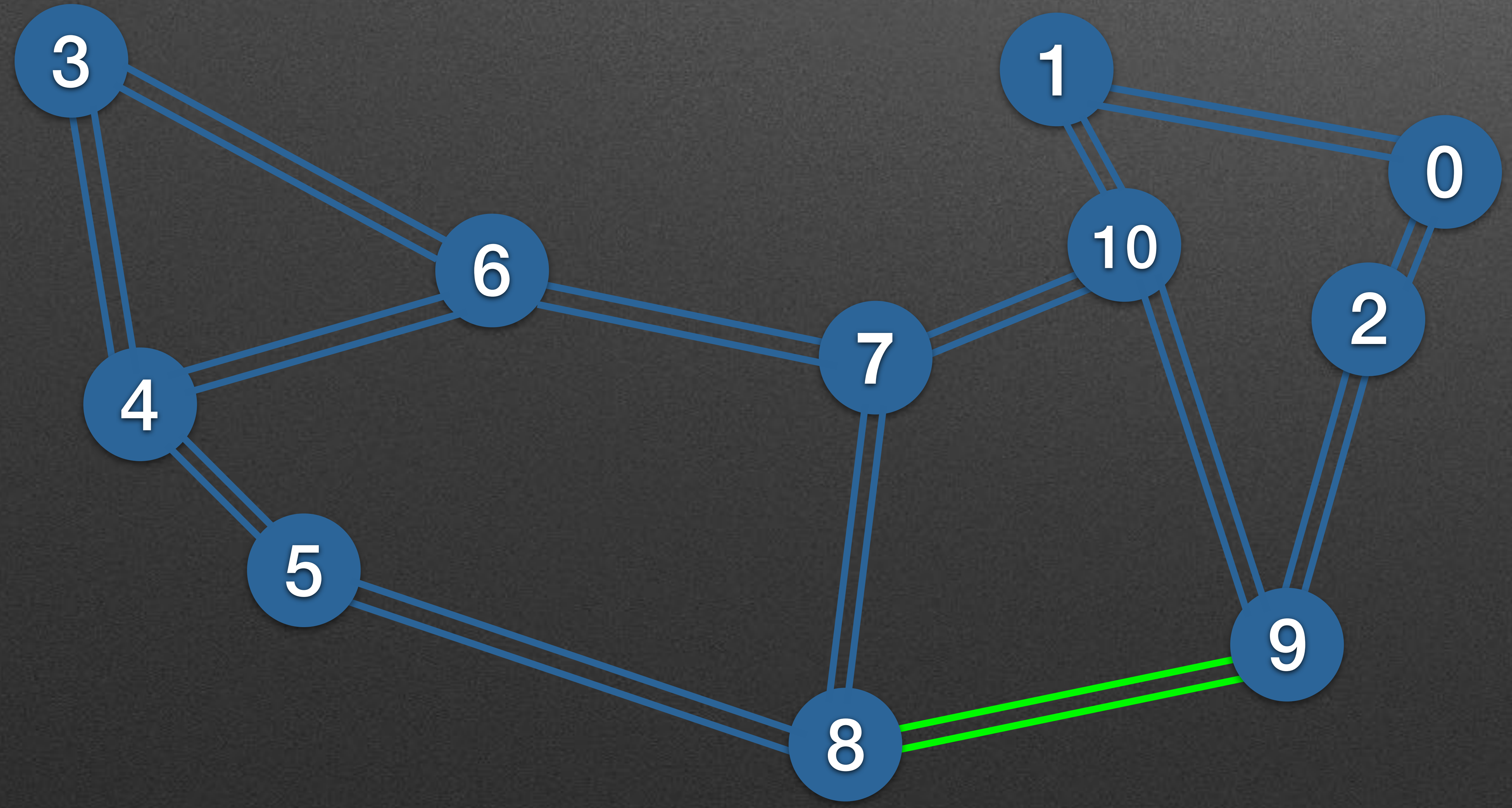
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274	(1, 10)	2.744
2	(2, 9), (10, 1)	1.274	(2, 9)	5.488
3	(9, 8), (10, 7)	1.217		

Capacity augmentation: Abilene

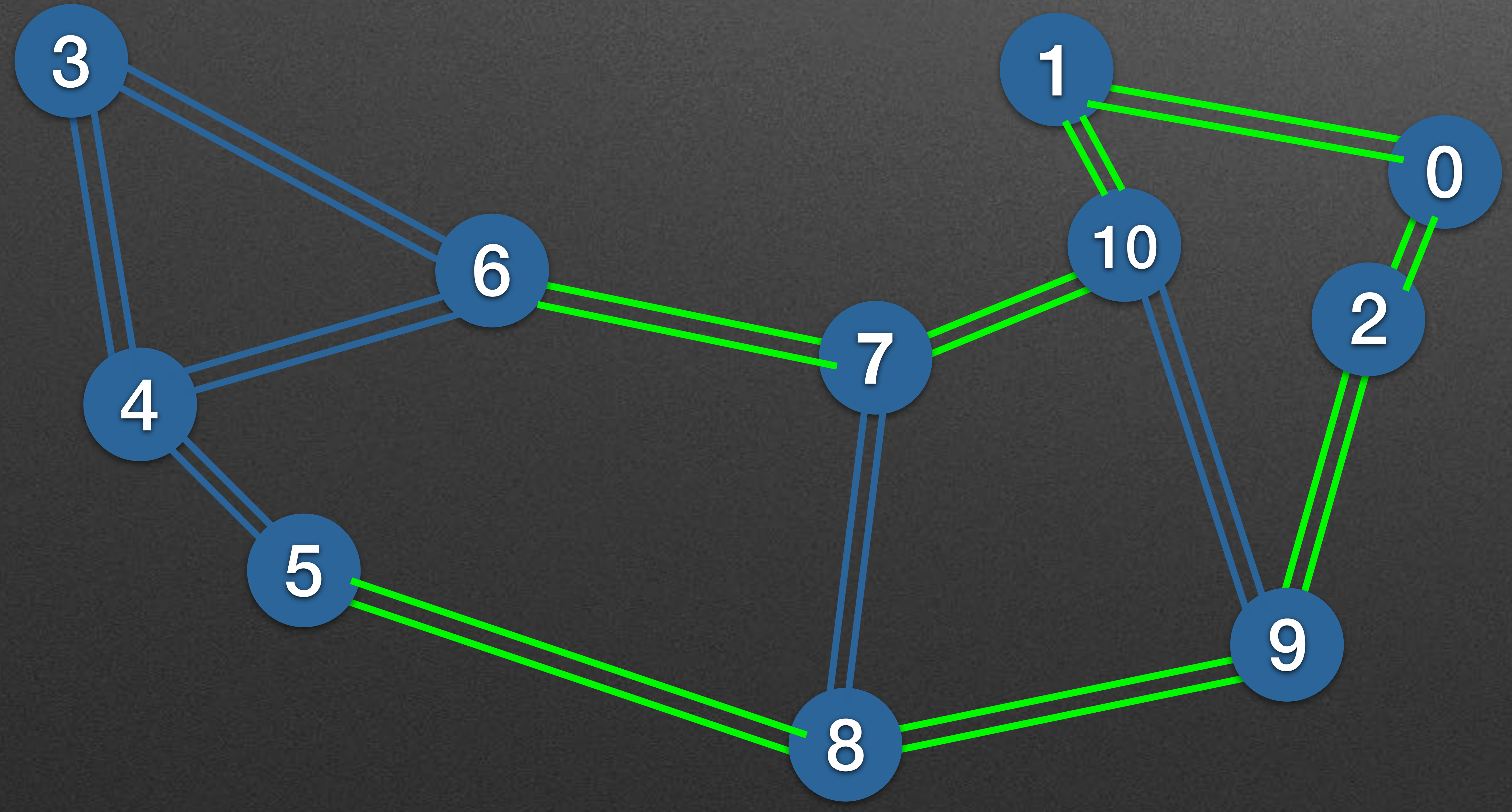
- Validate if MLU <= 1.
- If not, run augmentation LP with counter examples



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274	(1, 10)	2.744
2	(2, 9), (10, 1)	1.274	(2, 9)	5.488
3	(9, 8), (10, 7)	1.217	(9, 8)	7.653

Capacity augmentation: Abilene

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



Step	Counter examples	MLU	Links Augmented	Total new capacity (Gbps)
1	(1, 10), (2, 9)	1.274	(1, 10)	2.744
2	(2, 9), (1, 10)	1.274	(2, 9)	5.488
3	(9, 8), (10, 7)	1.217	(9, 8)	7.653
4	(10, 7), (9, 8)	1.217	(10, 7)	9.818
5	(0, 2), (1, 10)	1.192	(0, 2)	11.743
6	(1, 0), (1, 10)	1.071	(1, 0)	12.452
7	(7, 6), (8, 5)	1.006	(7, 6)	12.509
8	(8, 5), (7, 6)	1.006	(8, 5)	12.566
9	—	1.000	—	—

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?