Priority-Based Progressive Service Restoration After Massive Network Disruption

Injung Kim[1]; Novella Bartolini[2]; Hana Khamfroush[1]; Thomas La Porta[1]
[1]Pennsylvania State University, Electrical Engineering and Computer Science Department
[2]Sapienza University of Rome, Computer Science Department

PENNSTATE
1 8 5 5

INSTITUTE FOR NETWORKING AND SECURITY RESEARCH

MOTIVATION

The vast scope of the disaster or large scale network failure may affect many network elements performance. Different recovery methods would restore the failed elements in different order and therefore, the amount of restored network capacity over time can be different for different approaches. Moreover, when the demand requests have some priority or deadline to be repaired, there are more chances to loose the satisfaction of demand requests.

Our work aims to find a priority-aware progressive service restoration sequence after massive network disruption, while taking into account real-world constraints such as network capacity, demand satisfaction priority, demand satisfaction deadline, recovery cost, etc. In this poster, we propose a priority-aware progressive service restoration algorithm that achieves the maximum demand capacity satisfaction over time.

PROJECT OBJECTIVES

Given knowledge of network status, obtain the priority-aware optimal recovery sequence that maximizes demand satisfaction over time by either

- Repairing the broken network elements, or
- Building the new network elements

Constraints

- Demand Priority Group
- Demand Capacity (DC)
- Demand Pair (DP)
- Supply Capacity (SC)
- Repairing Resources
- Repairing Priority/Deadline
- Repairing Time and Cost

RECOVERY CANDIDATES & SEQUENCE

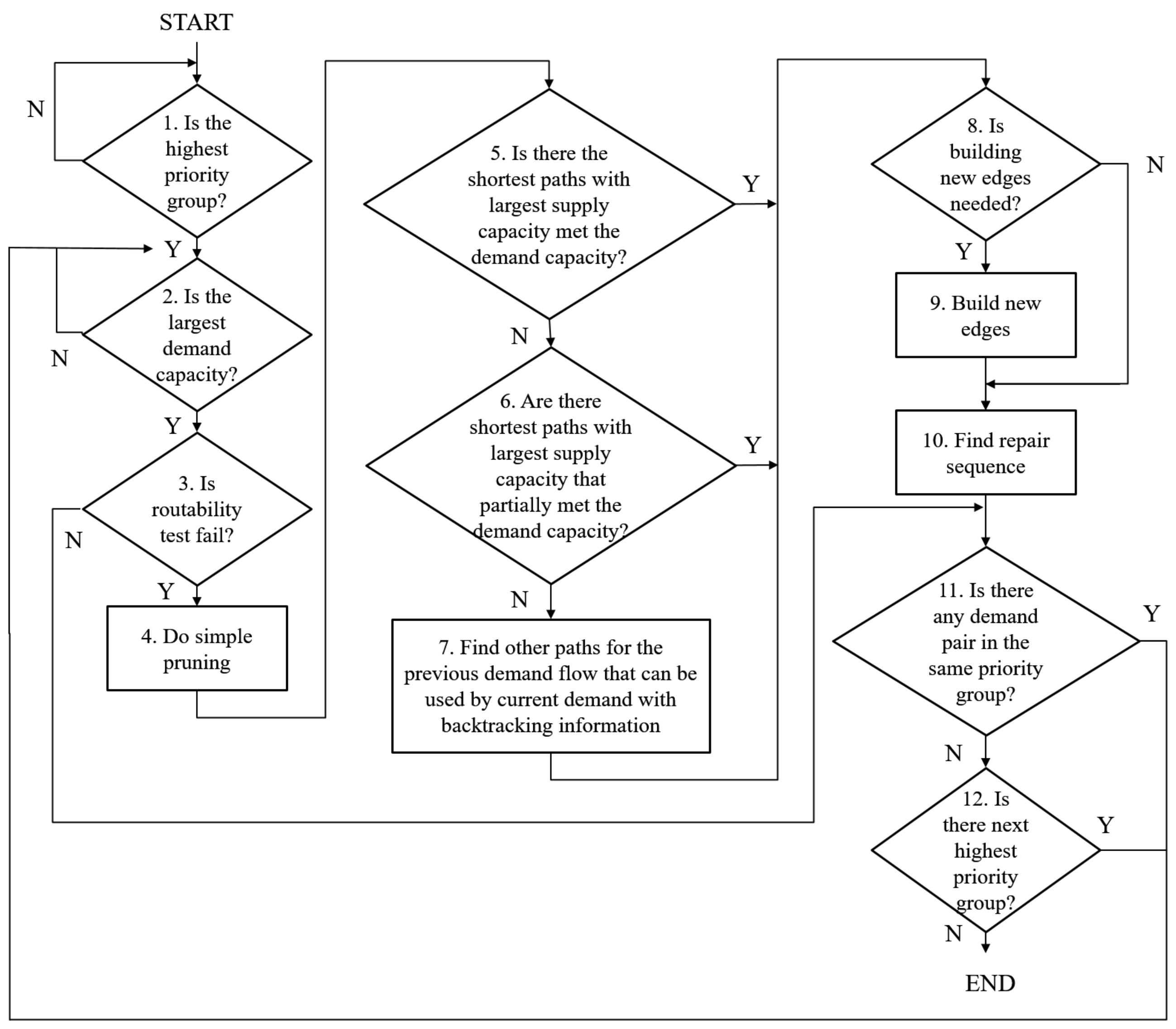
The way to find set of recovery candidates:

1. One shortest path with largest supply capacity that met the demand capacity
2. One or more shortest paths with largest supply capacity that partially met the demand capacity
3. Backtracking possible paths for current demand requests find other paths for the previous demand pairs, then current path can be used for current demand requests

Finding repair sequence (order is flexible by needs) :

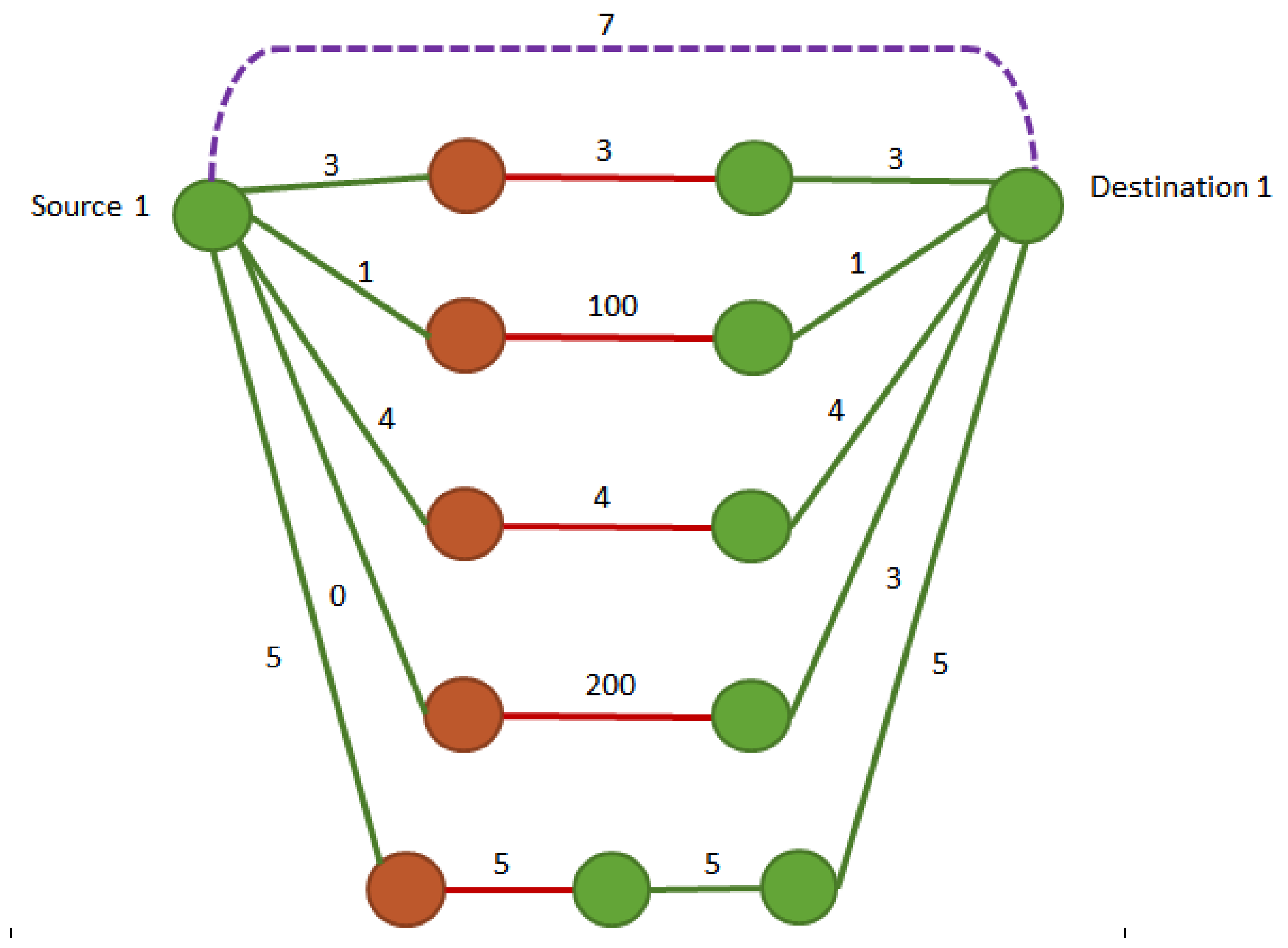
- Higher capacity
- Cheaper cost
- Shorter deadline
- Higher commonality

ALGORITHM

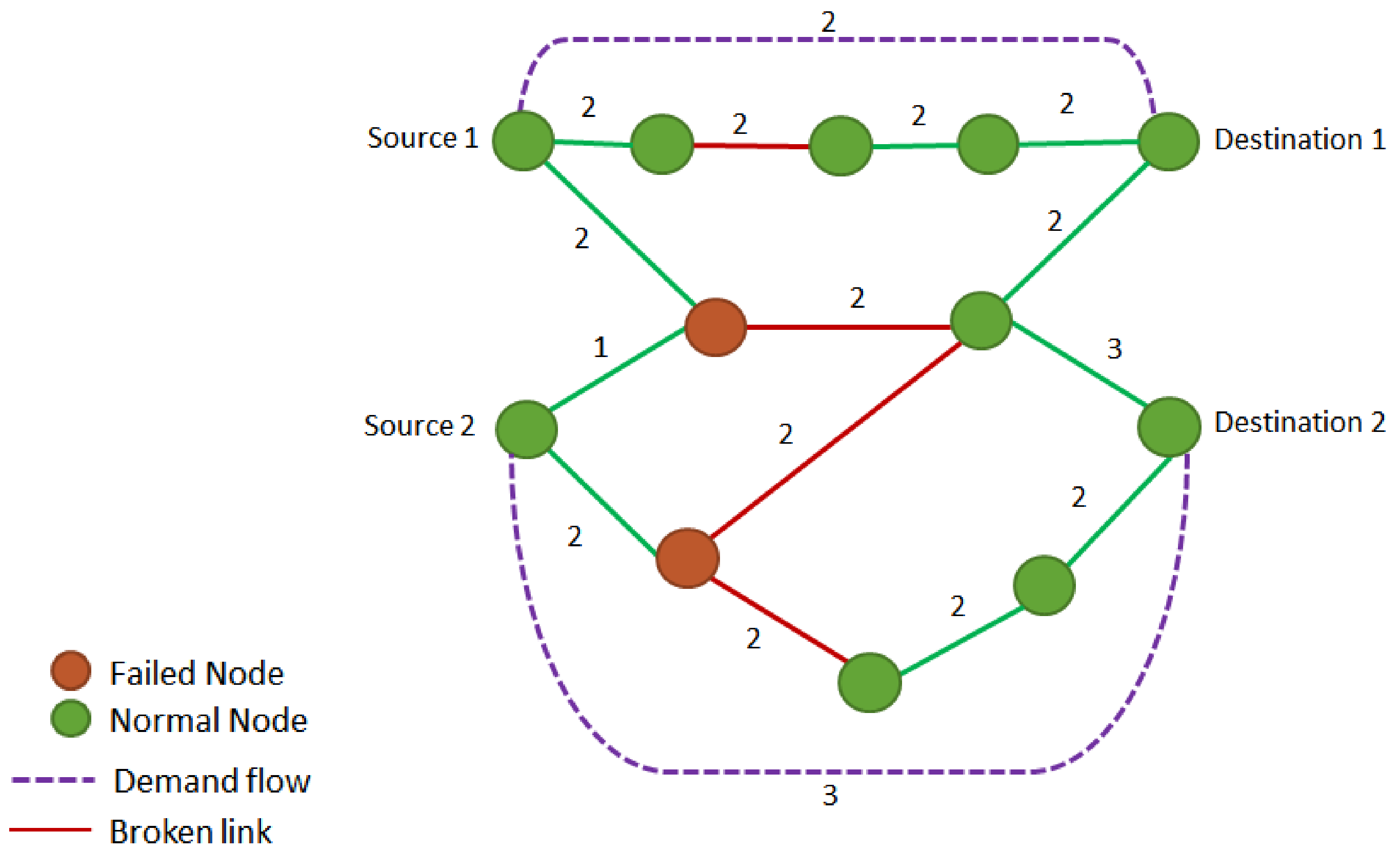


START
1. Is the highest priority group?
2. Is the largest demand capacity?
3. Is routability test fail?
4. Do simple pruning
5. Is there the shortest paths with largest supply capacity met the demand capacity?
6. Are there shortest paths with largest supply capacity that partially met the demand capacity?
7. Find other paths for the previous demand flow that can be used by current demand with backtracking information
8. Is building new edges needed?
9. Build new edges
10. Find repair sequence
11. Is there any demand pair in the same priority group?
12. Is there next highest priority group?
END

RECOVERY CANDIDATE CASES



Find Recovery Candidates Sets

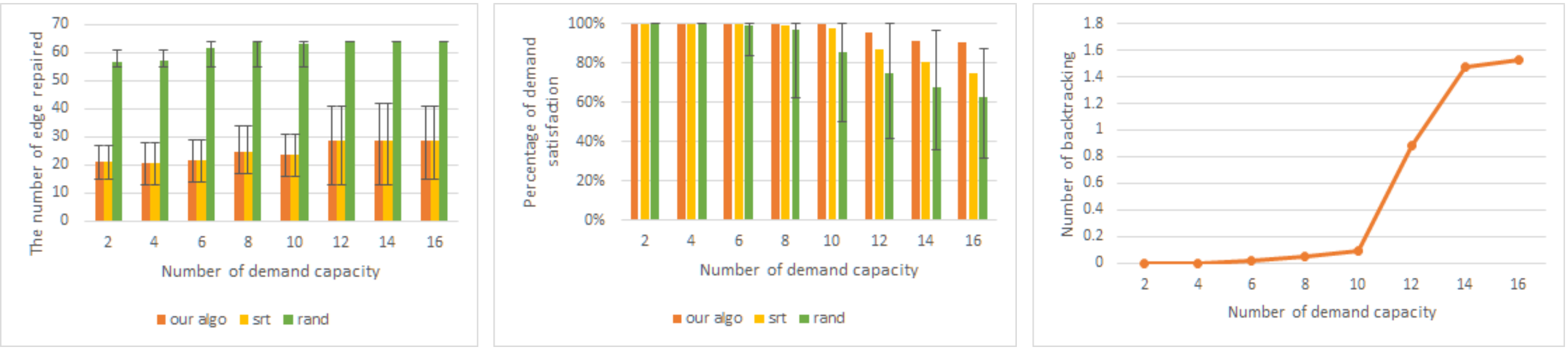


Find Recovery Candidates Sets with Backtracking

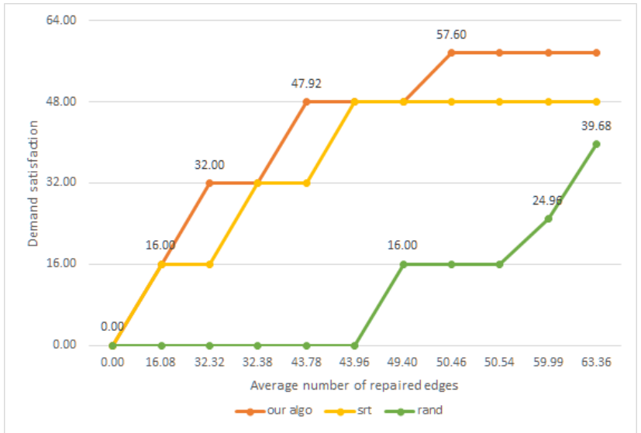
EXPERIMENTS

Bell-Canada Topology (48 nodes, 64 edges)

1. Unify Demand Intensity

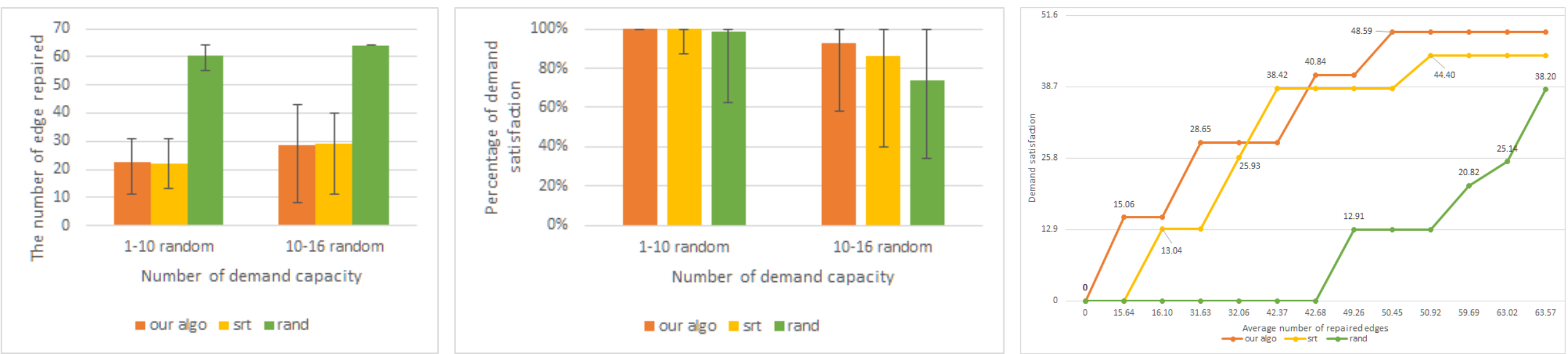


(SC:20, DC:2, 4, 8, 10, 12, 14, 16, DP: 4)

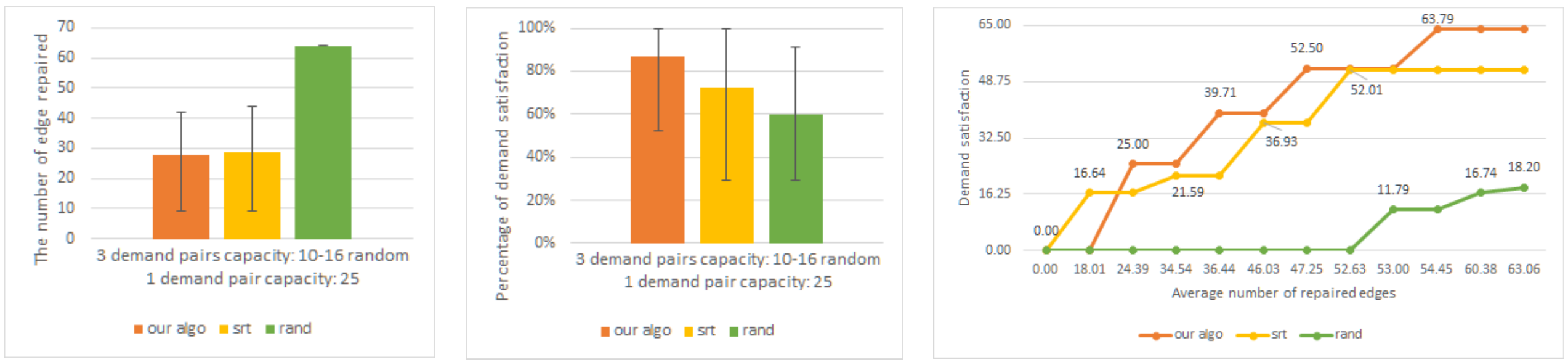


(SC20, DC: 16, DP: 4)

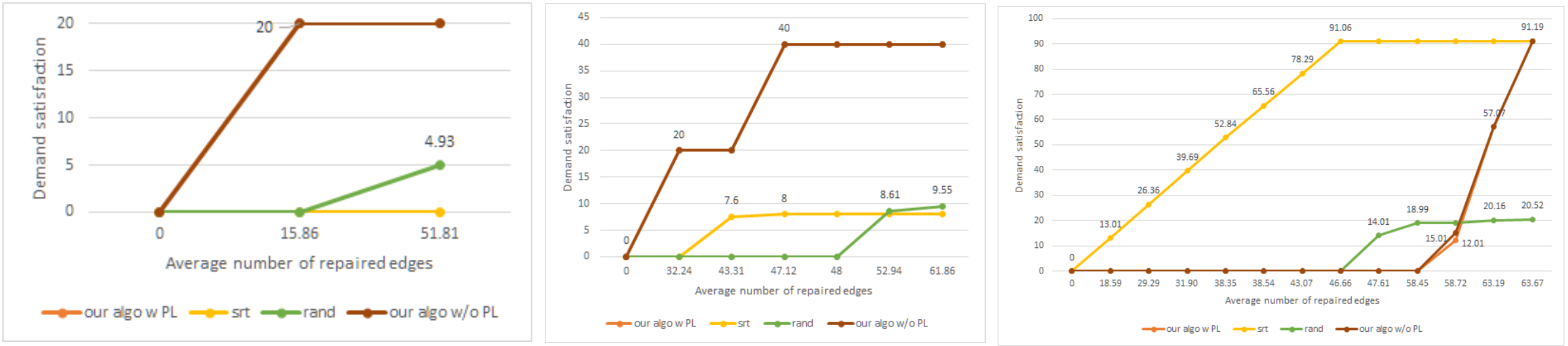
2. Heterogeneous Demand Intensity (SC:20, DC:1-10 (random), 10-16(random), DP: 4)



3. Heterogeneous Demand Intensity with one Large Capacity (SC:20, DC:1-10 (random) for 3 DPs & 25 for 1 DP, DP: 4)



4. Priority Level Locking (PL) vs. Unlocking Priority Level 1(10% DP), 2(20% DP), 3(70% DP) (SC:20, DC:20, 20& 10-16 (random), DP:10)



CONCLUSIONS

- We propose the priority based progressive service restoration to achieve resilient network that can handle large scale network disruption assuming demand priority groups.
- Our algorithm works based on the shortest paths and handled by largest capacity with backtracking if needed. The order of repairs is selected based on the priority class of the demand and the capacity that the repaired link can contribute to the path.
- The initial decision on routing some demands may affect the routability of the next demands, therefore, we allow the backtracking to re-route some of previously repaired demand pairs. We also allow building new wireless links with some additional costs.