Joint Backup Capacity Allocation and Embedding for Survivable Virtual Networks

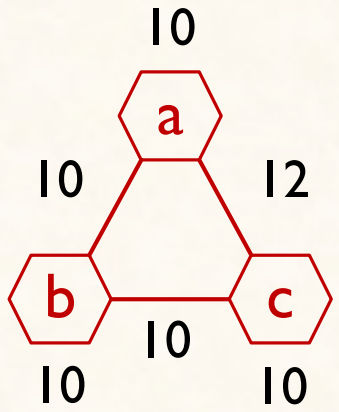
Nashid Shahriar,
Shihabur R. Chowdhury,
Reaz Ahmed, Aimal Khan,
Raouf Boutaba

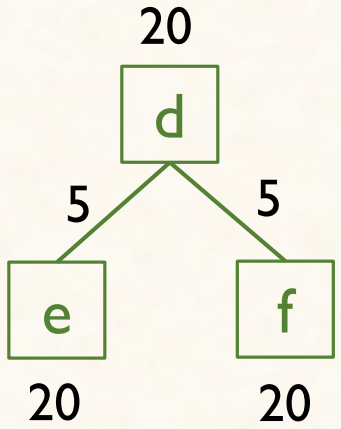
UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School
of Computer Science

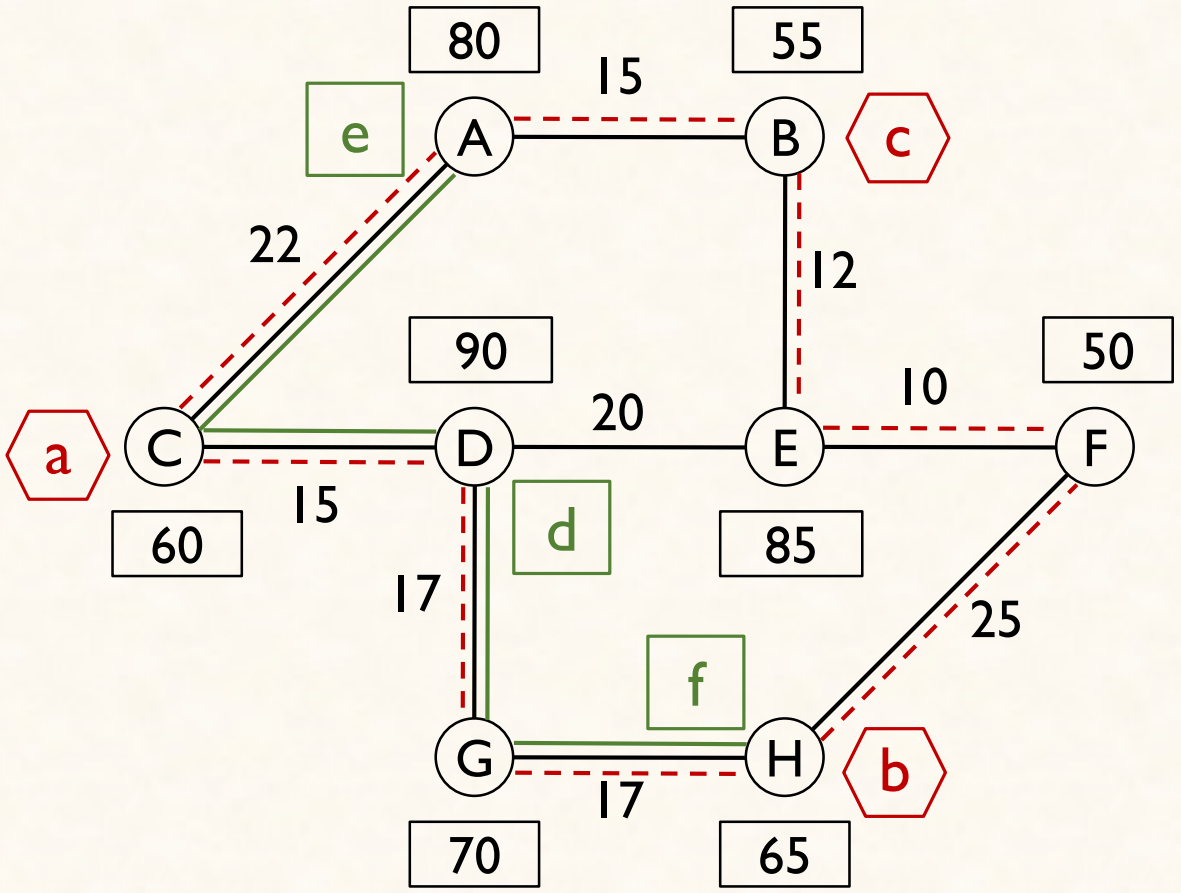
Jeebak Mitra,
Liu Liu

HUAWEI

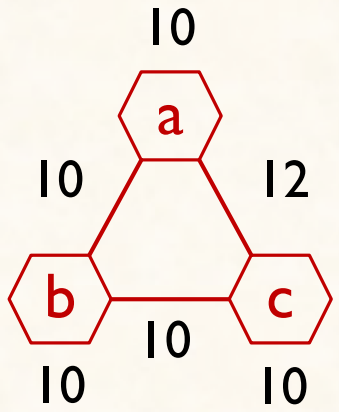
Virtual Network Embedding (VNE)

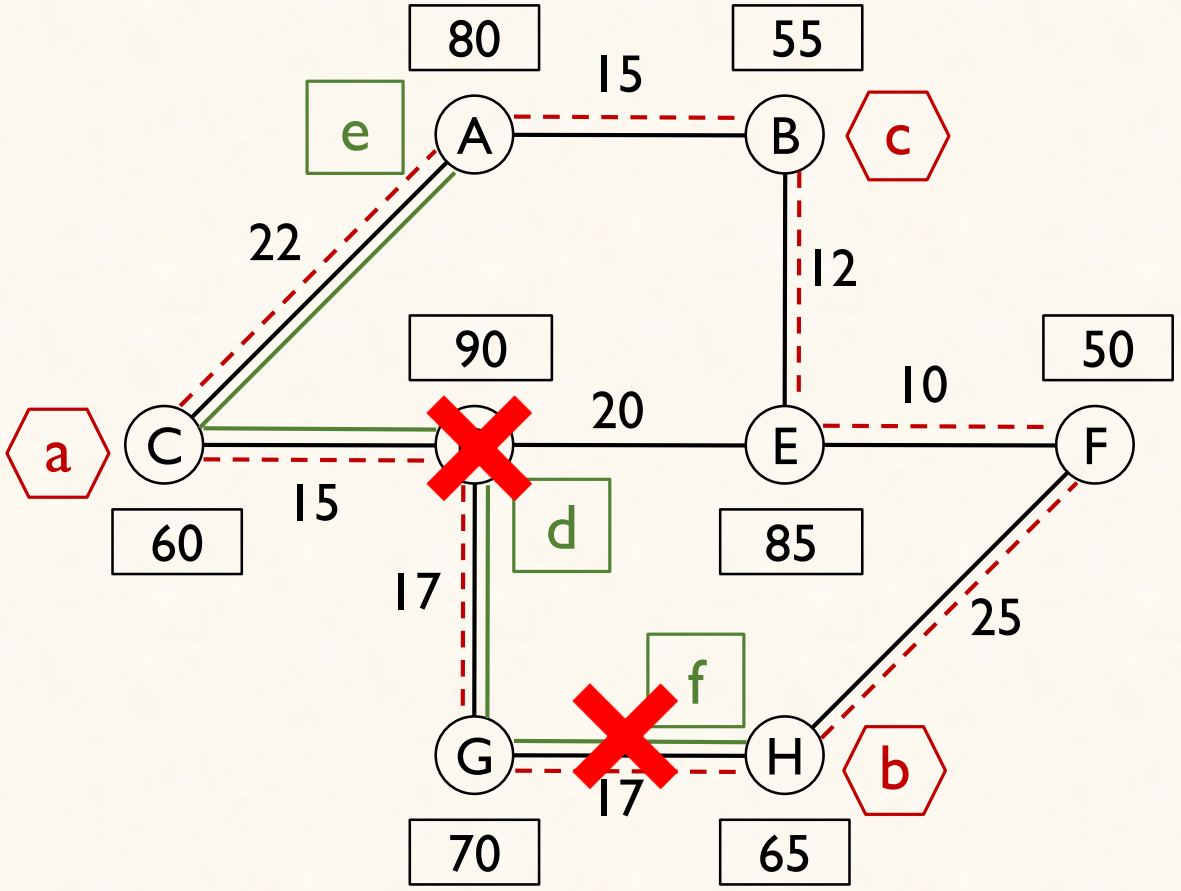






Virtual Network Embedding (VNE)





SVNE*: State-of-the-art

Proactive	Reactive
Pre-allocate backup resources in the substrate to guarantee VN QoS	*Reallocate* VNs completely or partially *after* a substrate failure

In general, SVNE approaches *keep the VN as is* and provision primary and backup resource in the substrate, possibly disjoint from each other.

* Survivable Virtual Network Embedding

Question:

How can we modify a VN to equip it with sufficient *spare capacity* and *embed* it so that the VN can *survive substrate link failures*?

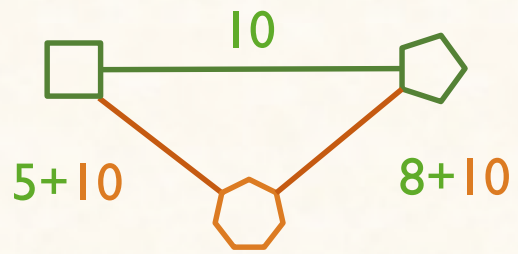
The Problem

Joint Backup Capacity Allocation and Embedding for Survivable Virtual Networks

The Problem

Joint Backup Capacity Allocation and Embedding for Survivable Virtual Networks

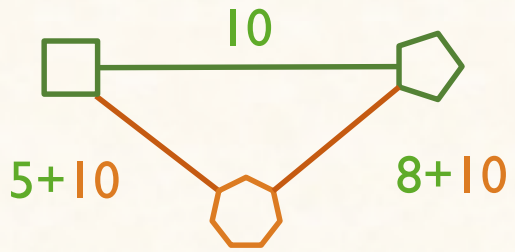
Allocate spare capacity on VLinks to act as backup of other VLinks



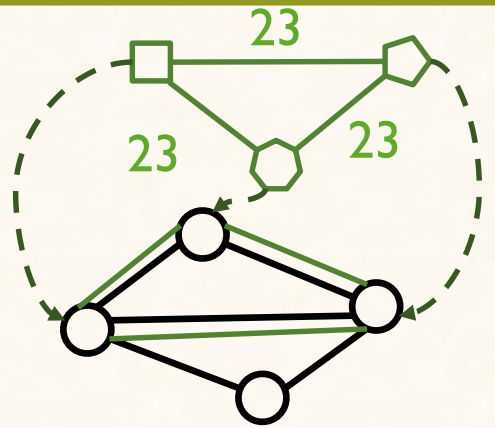
The Problem

Joint Backup Capacity Allocation and Embedding for Survivable Virtual Networks

Allocate spare capacity on VLinks to act as backup of other VLinks



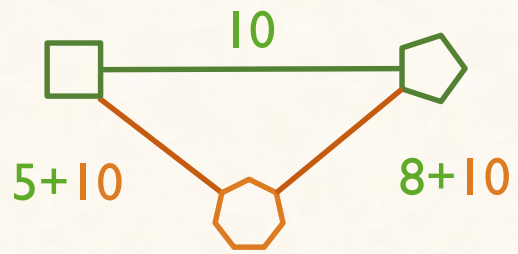
Embed VLinks disjointedly from their backup VPaths



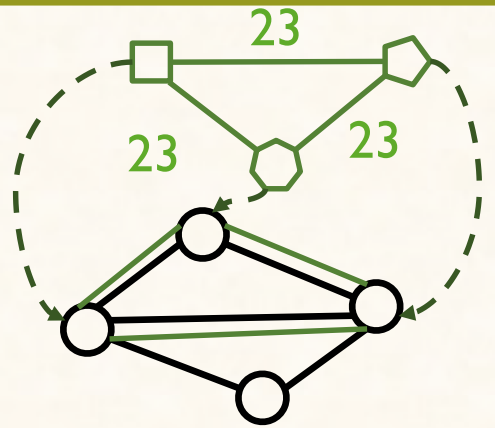
The Problem

Joint Backup Capacity Allocation and Embedding for Survivable Virtual Networks

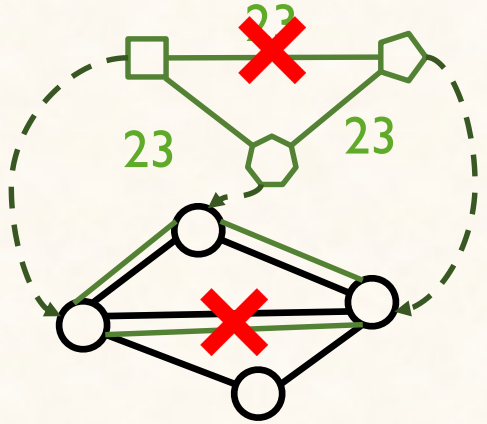
Allocate spare capacity on VLinks to act as backup of other VLinks



Embed VLinks disjointedly from their backup VPaths



Guaranteed bandwidth under single substrate link failure



Why Allocate Backup at the Level of VN?

Why Allocate Backup at the Level of VN?

Shift responsibility from infrastructure provider to service provider

Why Allocate Backup at the Level of VN?

Shift responsibility from infrastructure provider to service provider

Lesser cost of leasing compared to 1+1 backup (typical for operator networks)

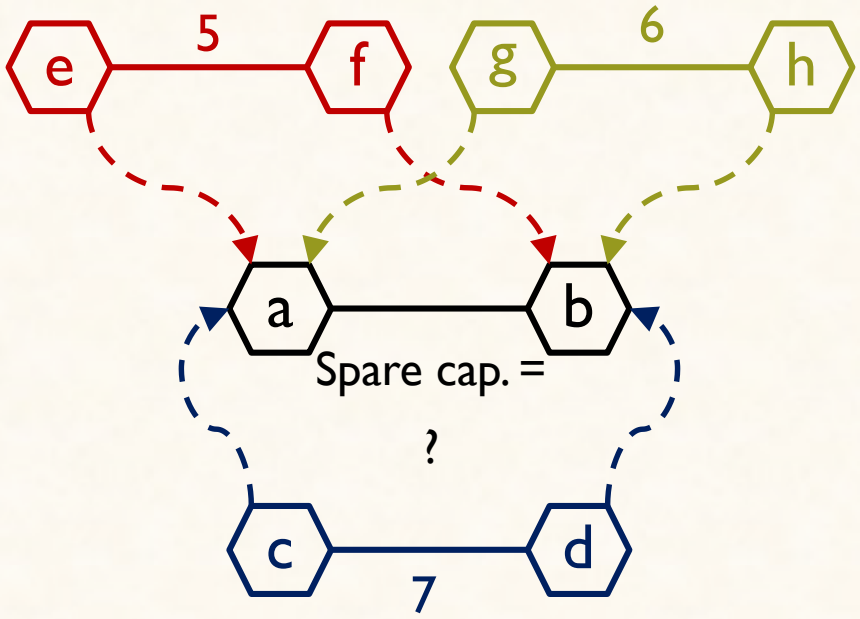
Why Allocate Backup at the Level of VN?

Shift responsibility from infrastructure provider to service provider

Lesser cost of leasing compared to 1+1 backup (typical for operator networks)

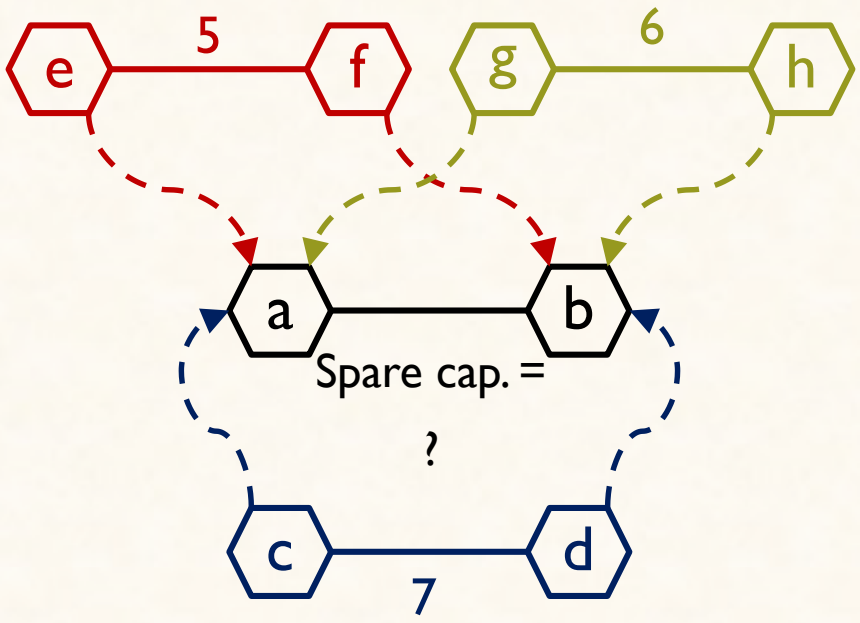
Potential for new service offerings

How to Assign Spare Capacity?



VLink (a, b) on backup paths for VLinks (e, f), (g, h), and (c, d)

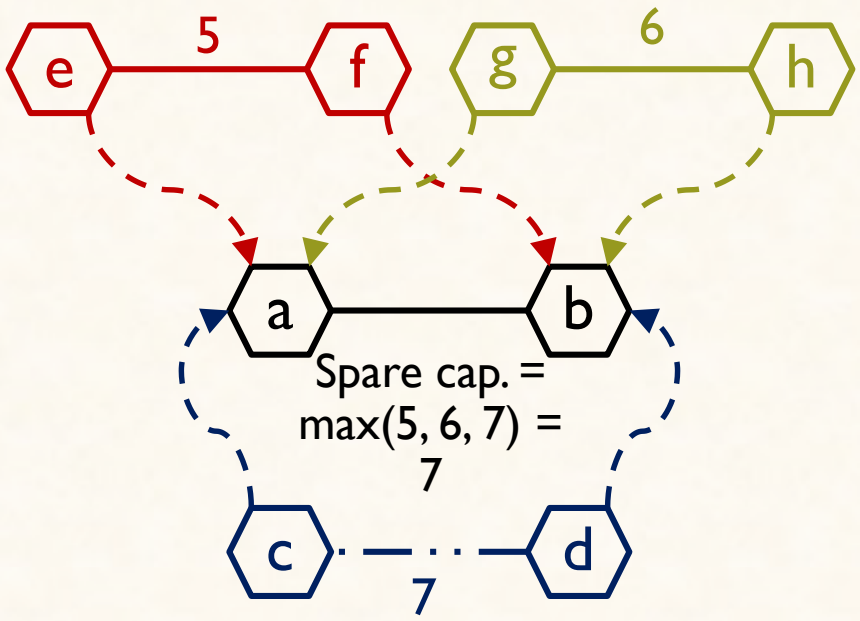
How to Assign Spare Capacity?



VLink (a, b) on backup paths for VLinks (e, f), (g, h), and (c, d)

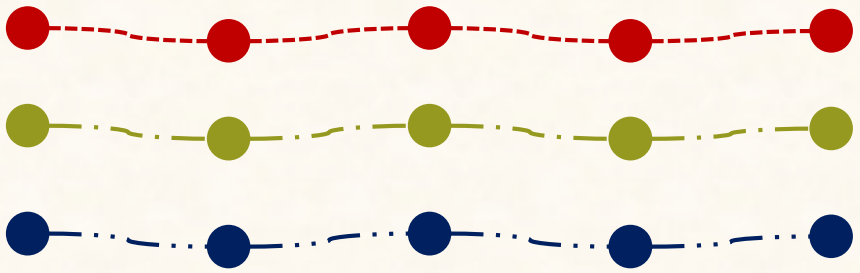
Depends on how the VLinks are embedded !

How to Assign Spare Capacity?

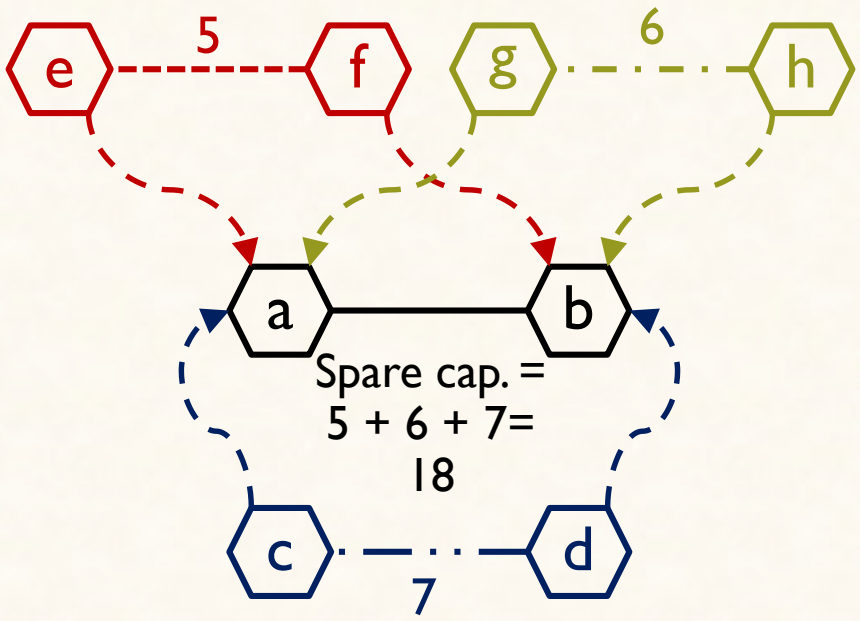


VLink (a, b) on backup paths for VLinks (e, f), (g, h), and (c, d)

Embedding Case – I:
All three are disojoint

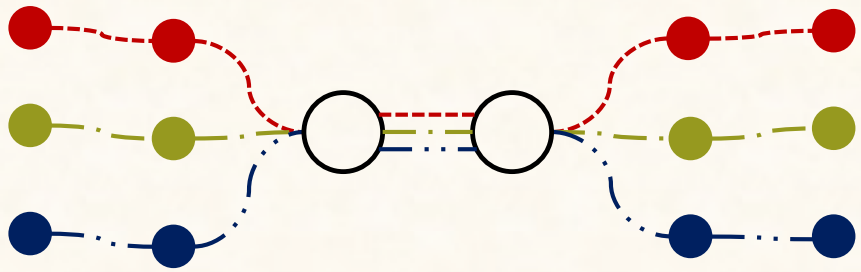


How to Assign Spare Capacity?

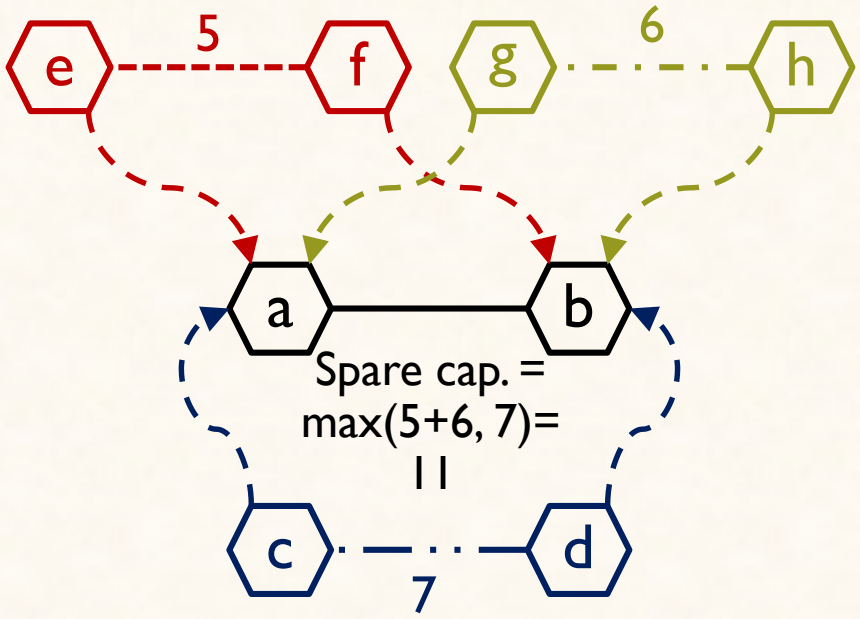


VLink (a, b) on backup paths for VLinks (e, f), (g, h), and (c, d)

Embedding Case – II:
None of them are disjoint

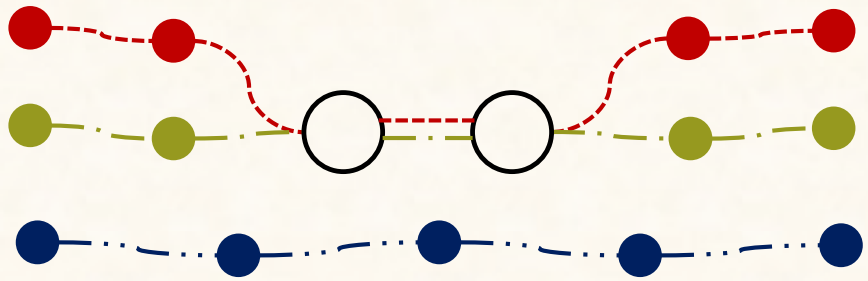


How to Assign Spare Capacity?

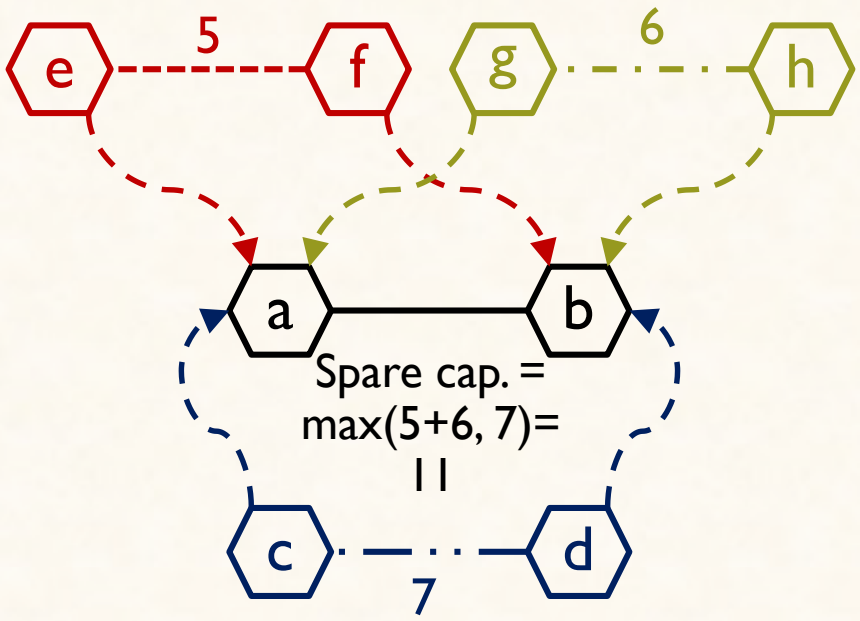


VLink (a, b) on backup paths for VLinks (e, f), (g, h), and (c, d)

Embedding Case – II:
Two disjoint groups



How to Assign Spare Capacity?



VLink (a, b) on backup paths for VLinks (e, f), (g, h), and (c, d)

Take-away

- Between different disjoint embedding groups *aka Shared Risk Group (SRG)*, spare capacity can be shared.
- Within an SRG, spare capacity is the *sum of all requirements*
- Total spare capacity is *maximum of required capacity of the SRGs*

Our Contribution

A suit of solutions

Optimal Solution

Formulated as an IQP and then transformed to an ILP* (NP-hard)

Heuristic

Three Step Heuristic: Estimate, Embed, Reconfigure

* Details is in the paper

Optimal Solution

Decision Variables

Assign *VLink to SRGs*, Assign *backup VPath to VLinks*, Allocate *spare capacity on VLinks*, Assign *VLink to SPath* and *VNode to SNode*

Quadratic Constraints

- SRG selection determines Spare capacity allocation.
- Spare capacity allocation determines bandwidth required on Slinks
- Together they yield quadratic constraint

IQP transformed to ILP*

Optimal Solution

Decision Variables

Assign VLink to SRGs, Assign backup VPath to VLinks, Allocate spare capacity on VLinks, Assign VLink to SPath and VNode to SNode

Quadratic Constraints

- SRG selection determines Spare capacity allocation.
- Spare capacity allocation determines bandwidth required on SLinks
- Together they yield quadratic constraint

IQP transformed to ILP*

Optimal Solution

Decision Variables

Assign VLink to SRGs, Assign backup VPath to VLinks, Allocate spare capacity on VLinks, Assign VLink to SPath and VNode to SNode

Quadratic Constraints

- SRG selection determines Spare capacity allocation.
- Spare capacity allocation determines bandwidth required on Slinks
- Together they yield quadratic constraint

IQP transformed to ILP*

* Details is in the paper

Heuristic Design: Challenges

Heuristic Design: Challenges

Spare capacity allocation depends on how VLinks are embedded

Heuristic Design: Challenges

Spare capacity allocation depends on how VLinks are embedded

Cannot embed VLinks without knowing total bandwidth requirement

Heuristic Design: A Chicken and Egg Problem

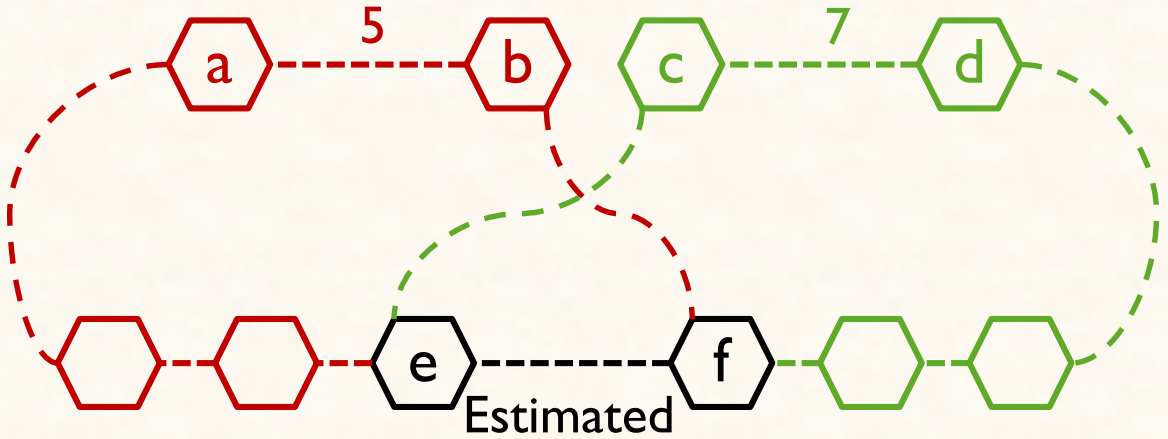
Spare capacity allocation depends on how VLinks are embedded

Cannot embed VLinks without knowing total bandwidth requirement

The Chicken and Egg Problem: Solution

Estimate, Embed, and Reconfigure

- Estimate backup VPaths by running weighted shortest path.
  - Adjust weights to reduce *spare capacity fragmentation*



Spare Cap. = 5 + 7 = 12

The Chicken and Egg Problem: Solution

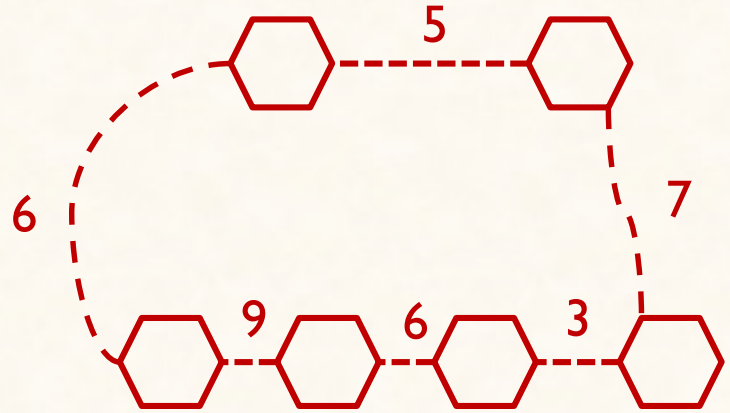
Estimate, Embed, and Reconfigure

- Embed VLinks with *estimated spare capacity*
  - Start embedding from most constrained (maximum degree) to least constrained VNode (minimum degree)
  - Use Dijkstra's shortest path algorithm with constraints to compute edge disjoint shortest paths for link embedding.

The Chicken and Egg Problem: Solution

Estimate, Embed, and Reconfigure

- Identify cycles in VN *s.t. no Vlink pair* on the cycle *shares an SLink in their embedding*.
  - Use the VLinks on such cycle as backup of each other



Required Spare Capacity:
max(5, 6, 9, 6, 3) = 9

Evaluation: Setup

- Heuristic compared with optimal solution
- Parameters
  - Substrate network
    - 20 – 90 nodes (small scale for comparison with optimal)
    - 500 and 1000 nodes (large scale)
    - Avg. node degree between 2 – 5
  - Virtual Network
    - 3 – 11 nodes (small scale)
    - 10 – 100 nodes (large scale)
    - Bandwidth demand = ~10% of substrate link bandwidth

Performance Highlights

Heuristic is within ***~1.2x*** *of optimal* on avg.

Heuristic was able to solve the problem for 50 node VN and 1000 node SN in under 8 seconds

Performance Highlights

Heuristic is within ***~1.2x*** *of optimal* on avg.

Heuristic was able to solve the problem for 50 node VN and 1000 node SN in under 8 seconds

Heuristic struggled to find solution for sparse substrate network (Node degree ~2)

Summary

We address a different form of survivability for VNs where the VN is equipped with the backup

We jointly optimize spare capacity allocation and VN embedding to obtain optimal solution

Our heuristic performs within **~1.2x** of the optimal (*empirically evaluated; not a theoretical bound*)

Questions?