Protecting Virtual Networks with DRONE

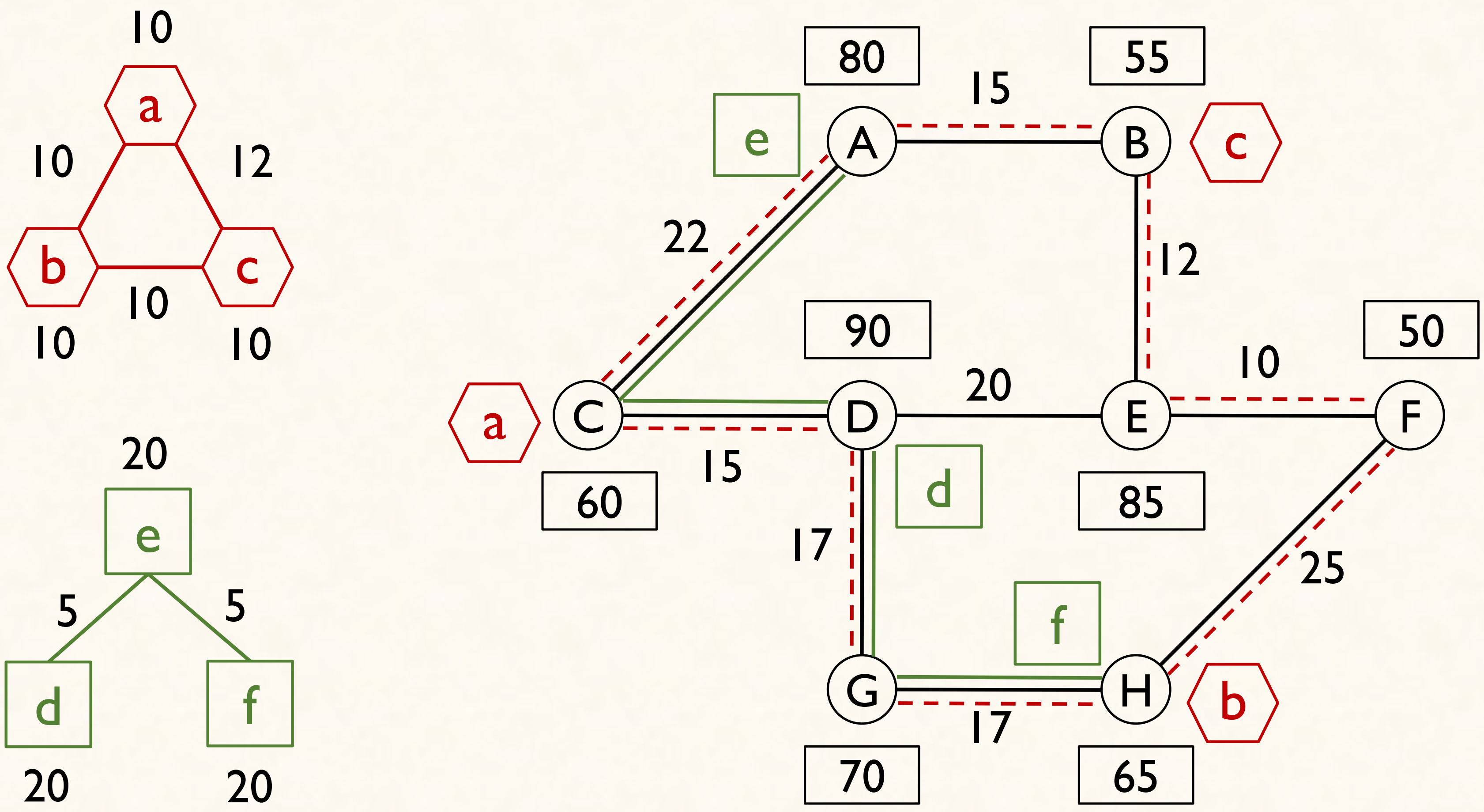
Shihabur R. Chowdhury,
Reaz Ahmed, Nashid Shahriar,
Md Mashrur Alam Khan,
Raouf Boutaba

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

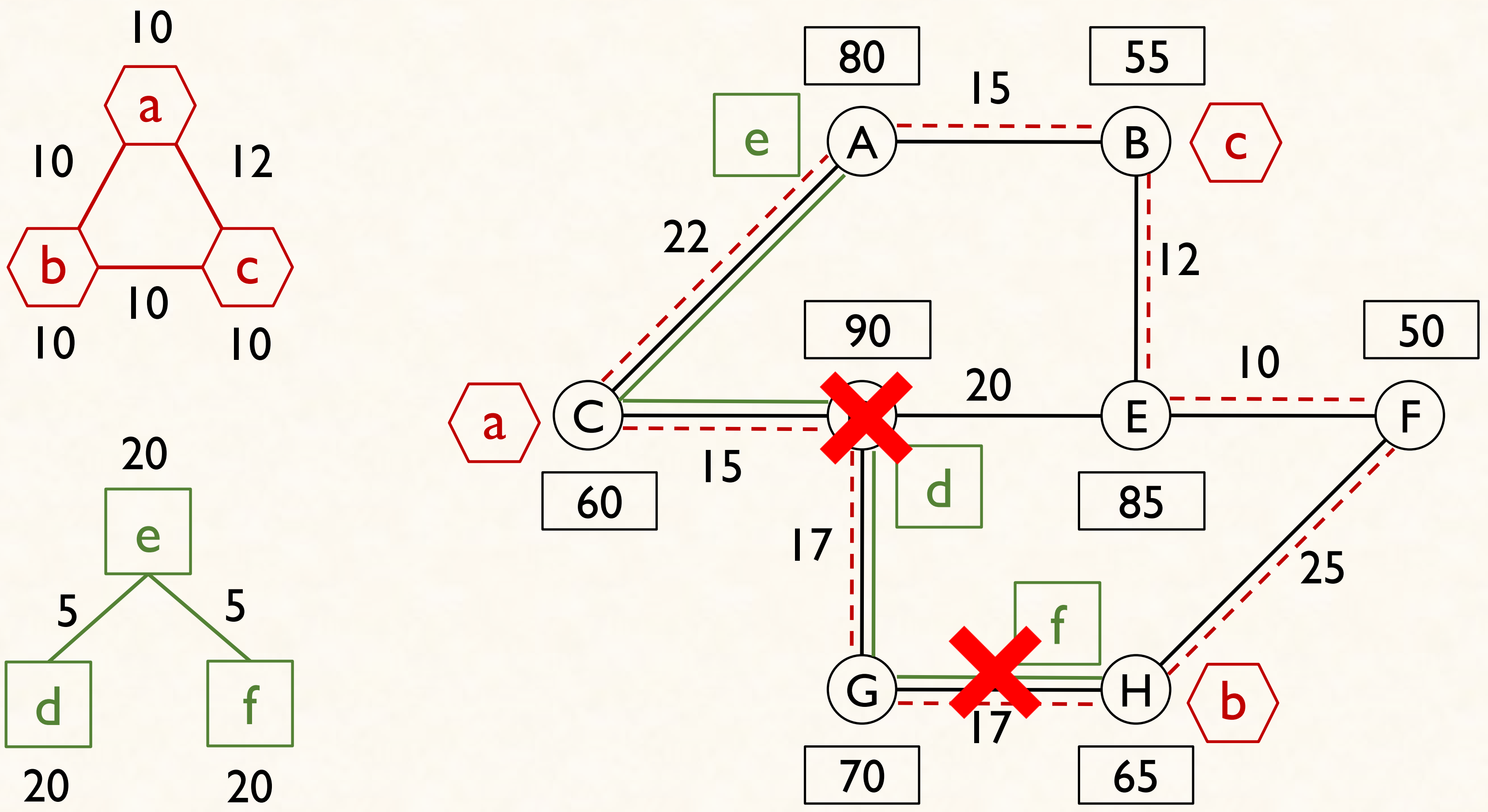
Jeebak Mitra,
Feng Zeng

HUAWEI

Virtual Network Embedding (VNE)



Virtual Network Embedding (VNE)



Transport Network Virtualization

Transport Networks traditionally provided end-to-end connectivity

New use cases with virtualization: Full fledged VNs for customers

Tight SLA: Recover within 50ms of failure

Key Question:

How to provide 1 + 1-protection to an entire virtual network in a resource efficient way?

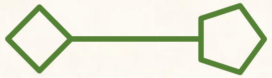
The Problem

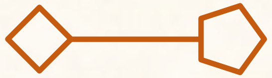
1 + 1- Protected Virtual Network Embedding
(1 + 1- ProViNE)

The Problem

1 + 1- Protected Virtual Network Embedding
(1 + 1- ProViNE)

Dedicated protection for each virtual resource

Primary VN: 

Backup VN: 

The Problem

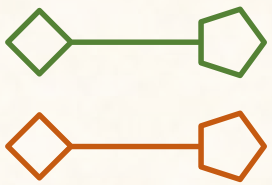
1 + 1- Protected Virtual Network Embedding (1 + 1- ProViNE)

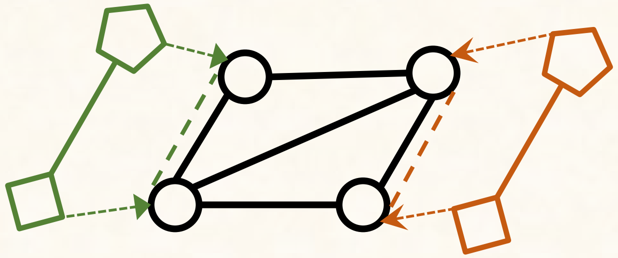
Dedicated protection for each virtual resource

Disjoint primary and backup embedding

Primary VN:

Backup VN:





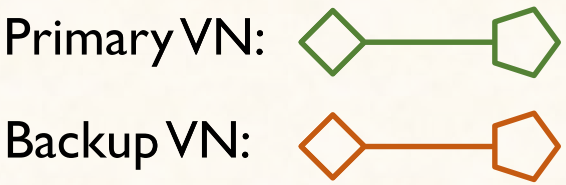
The Problem

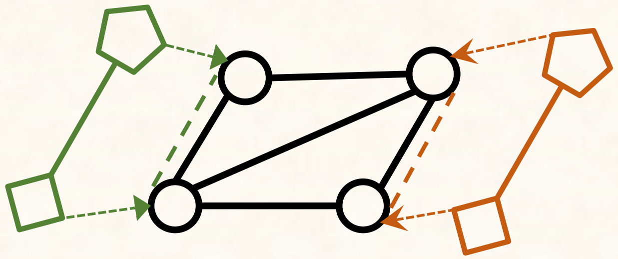
1 + 1- Protected Virtual Network Embedding
(1 + 1- ProViNE)

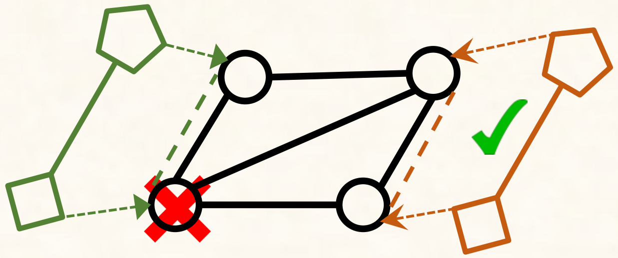
Dedicated protection for each virtual resource

Disjoint primary and backup embedding

Guaranteed Service under single physical node failure

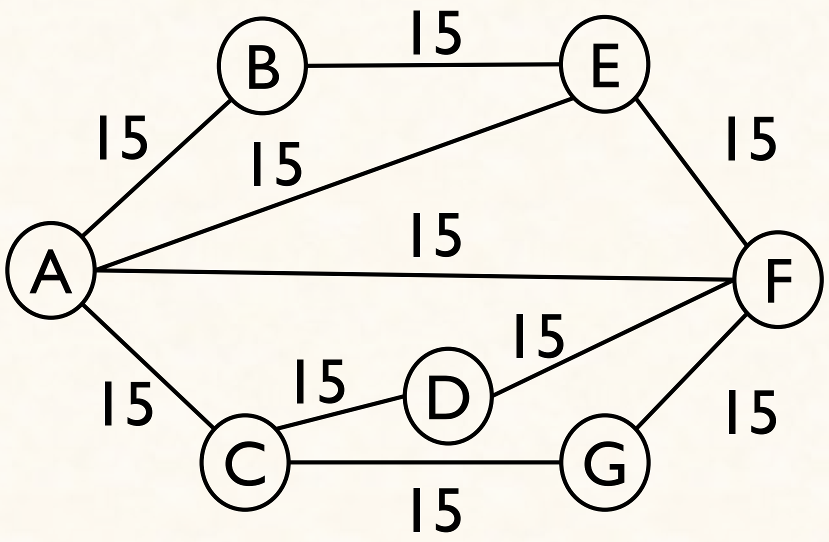




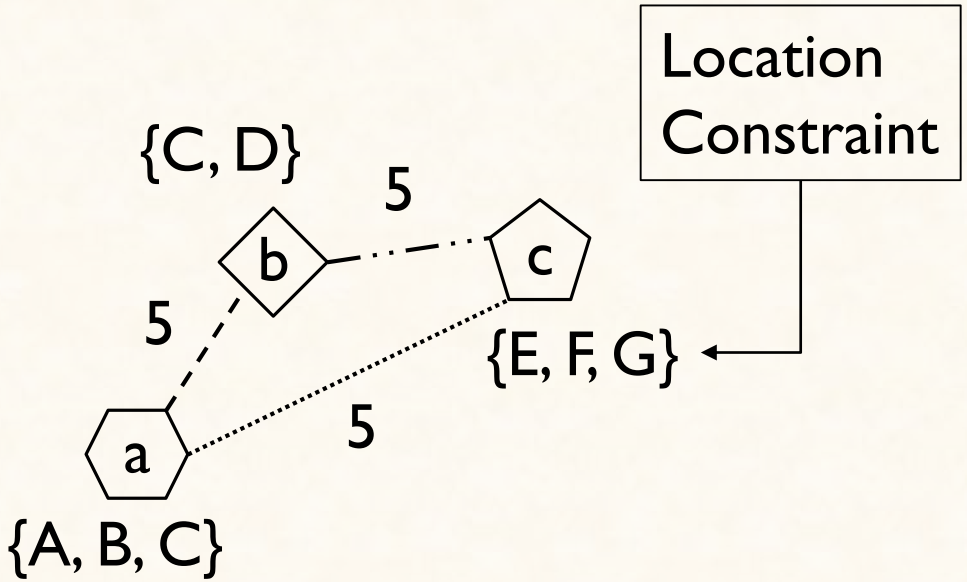


1+1-ProViNE: Example

Given



Physical Network
(G)

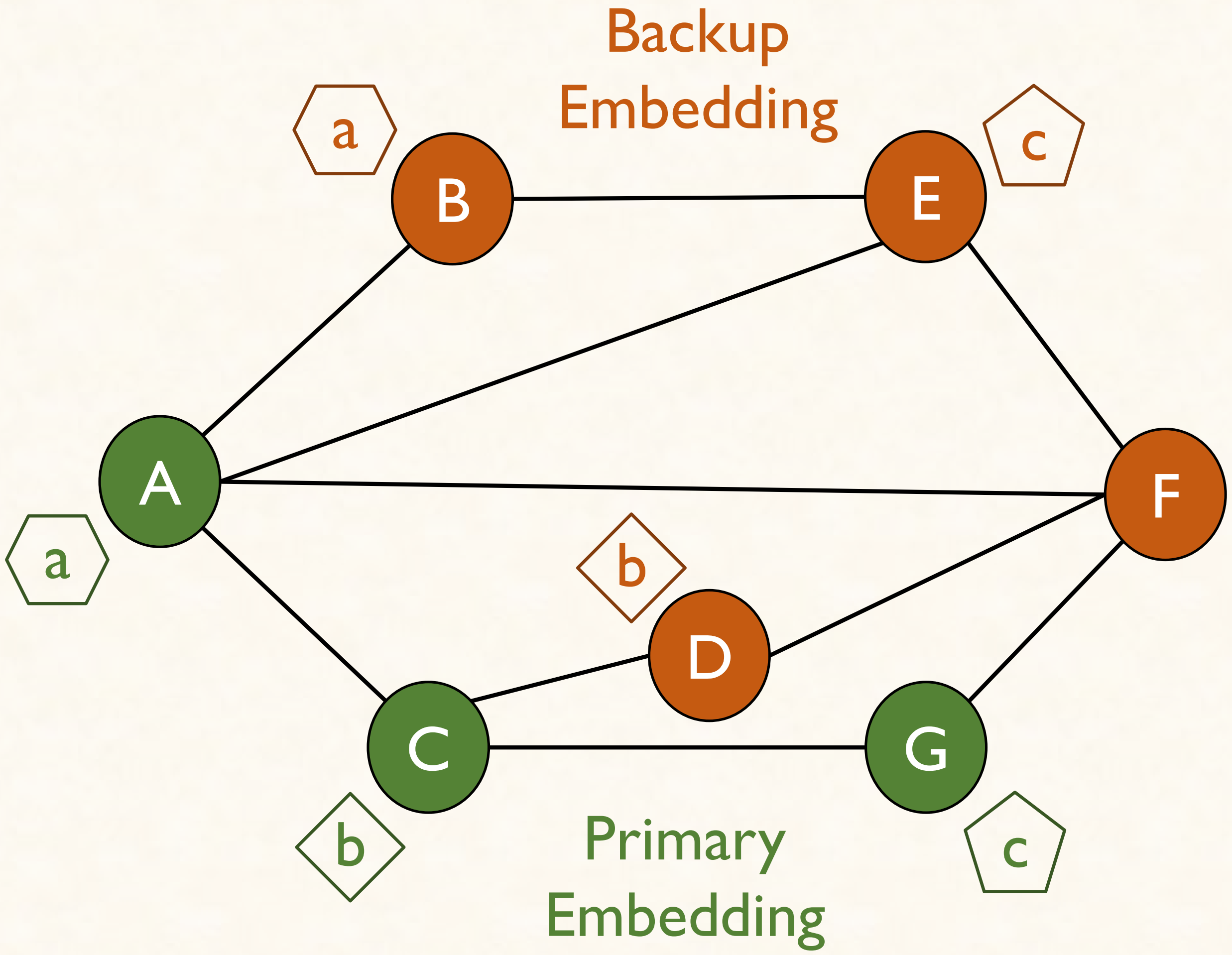


Virtual Network
(G')

1+1-ProViNE: Example

Find two disjoint embedding of G' on G

Two disjoint embedding of each virtual node

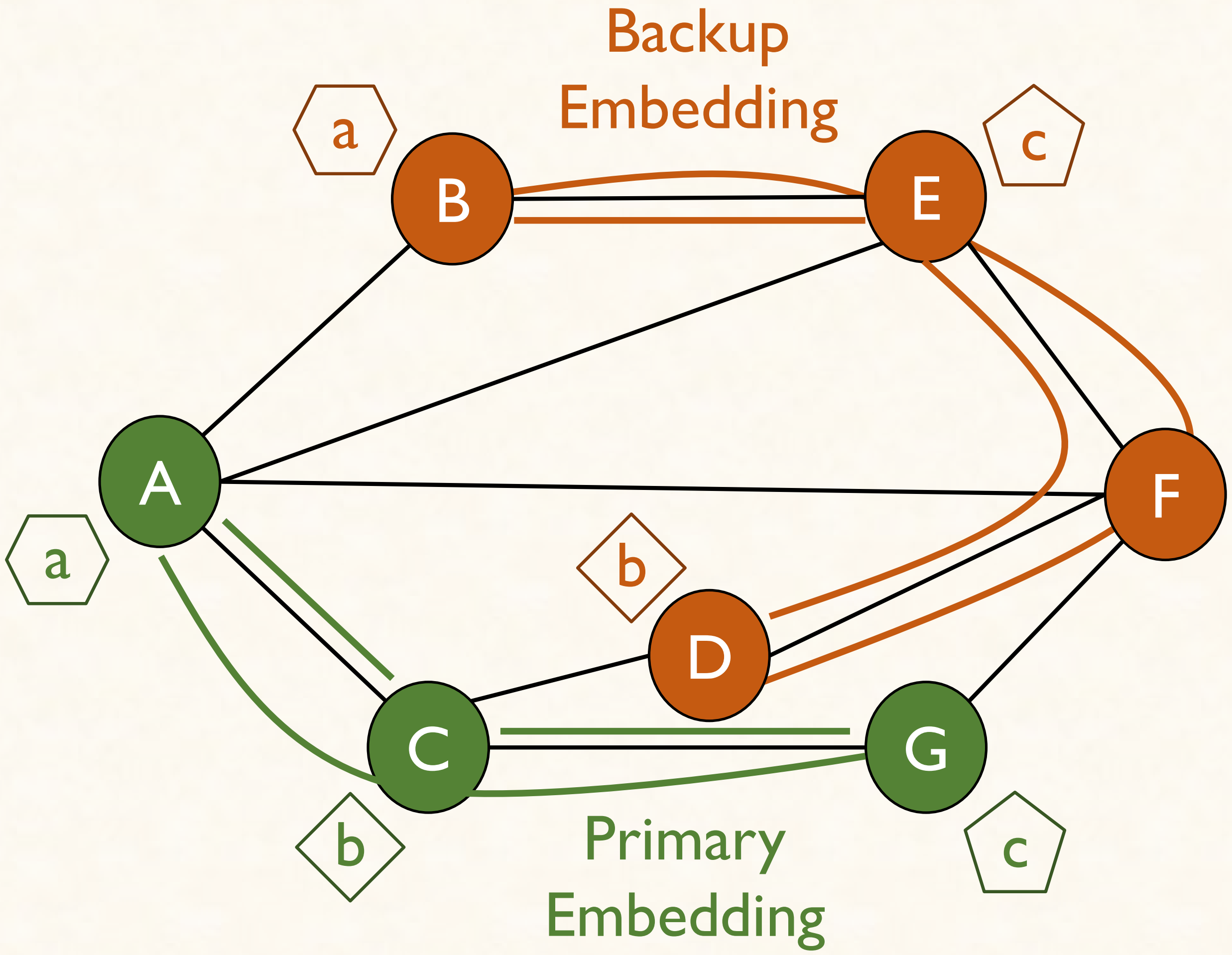


1+1-ProViNE: Example

Find two disjoint embedding of G’ on G

Two disjoint embedding of each virtual node

Two disjoint embedding of each virtual link



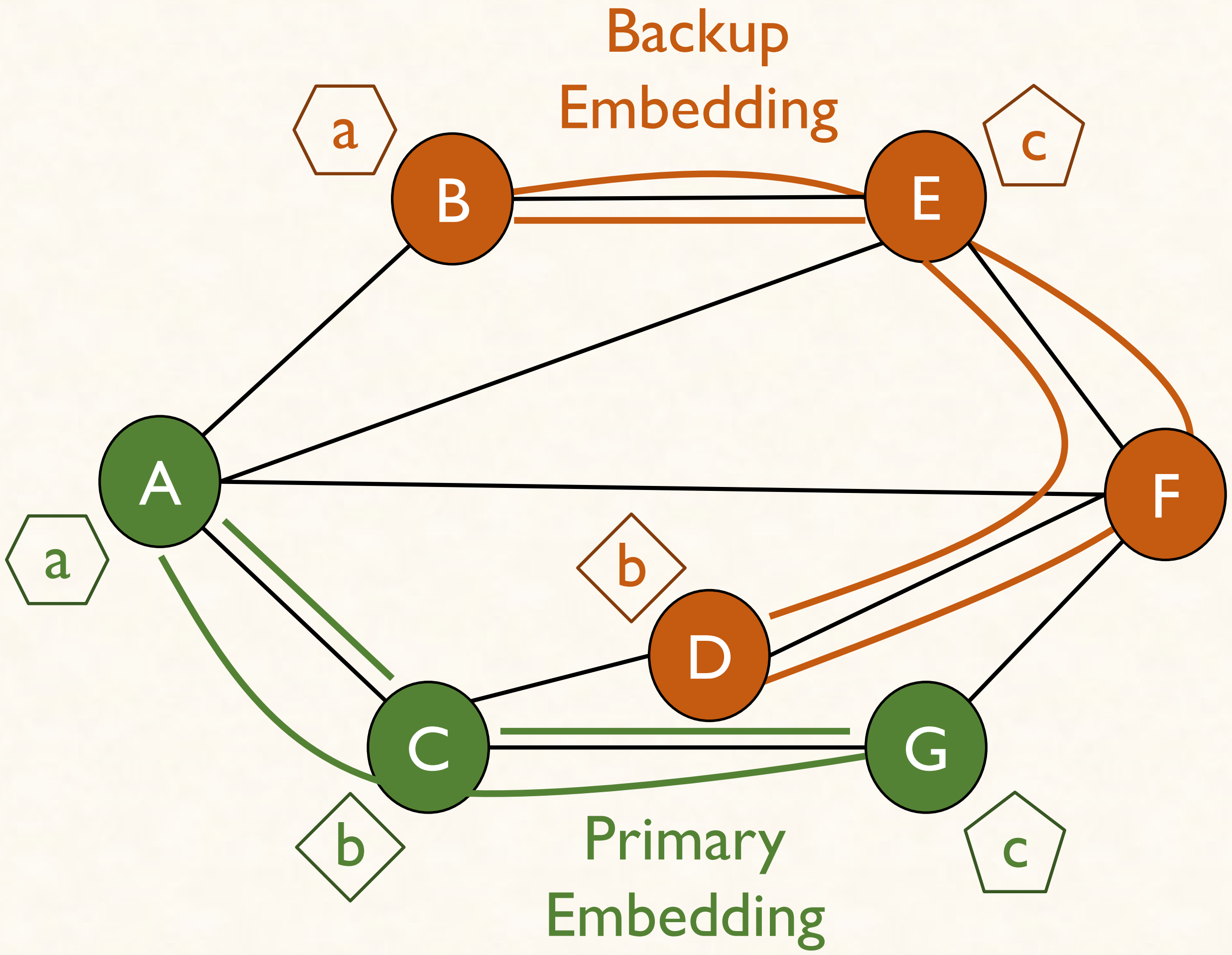
1+1-ProViNE: Example

Find two disjoint embedding of G’ on G

Two disjoint embedding of each virtual node

Two disjoint embedding of each virtual link

Objective: Minimize total bandwidth cost



Our Proposal

Dedicated Protection for Virtual Network Embedding (DRONE)

A suit of solutions to 1 + 1-ProViNE

OPT-DRONE	FAST-DRONE
ILP-based optimal solution	Fast heuristic

State-of-the-art

PAR*

Quadratic Integer Program

Maximizes Acceptance Ratio

First fit for node and Suurballe's alg. for link embedding, respectively

DRONE

Integer Linear Program

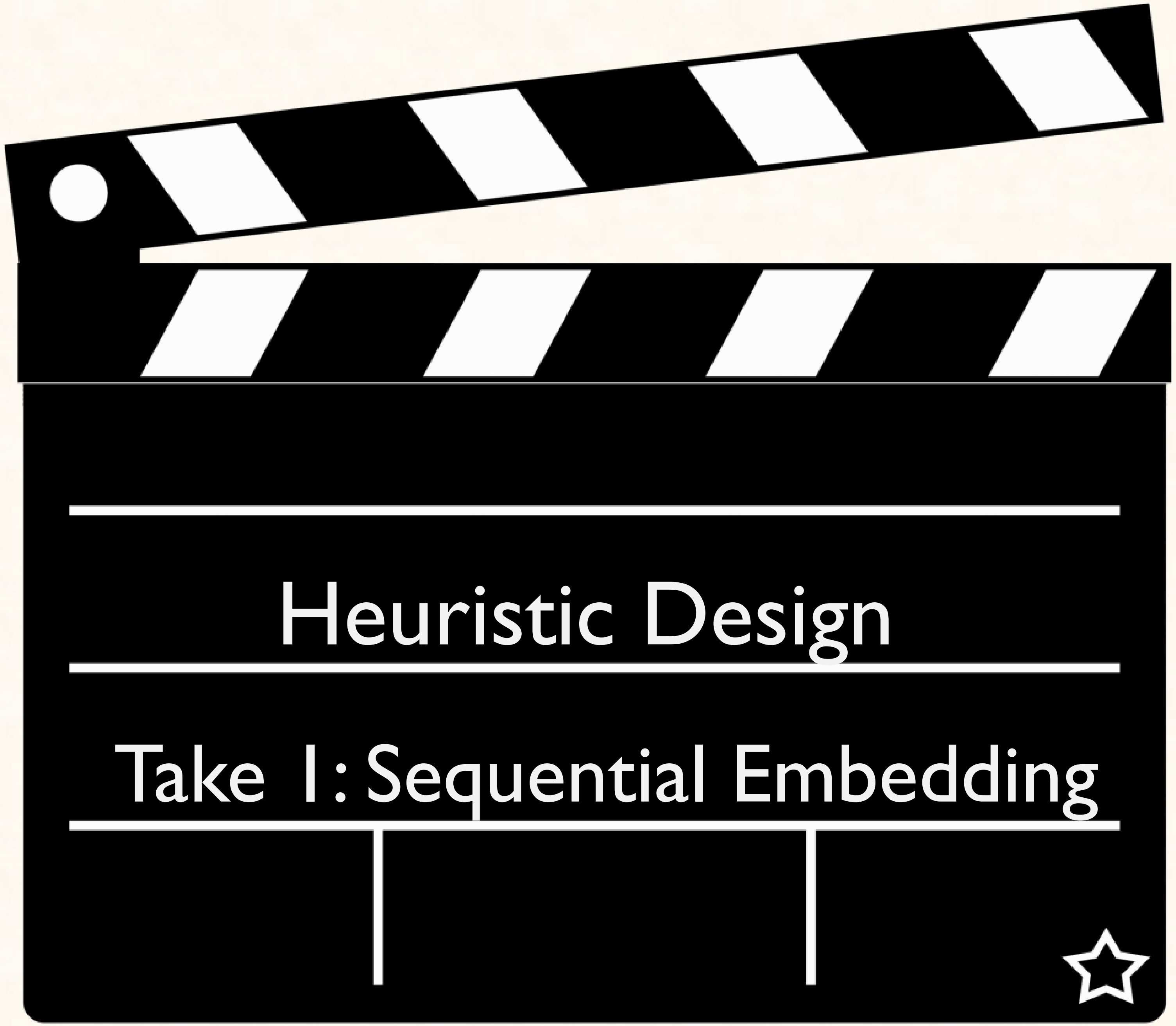
Minimizes Bandwidth Cost

Graph partition based heuristic

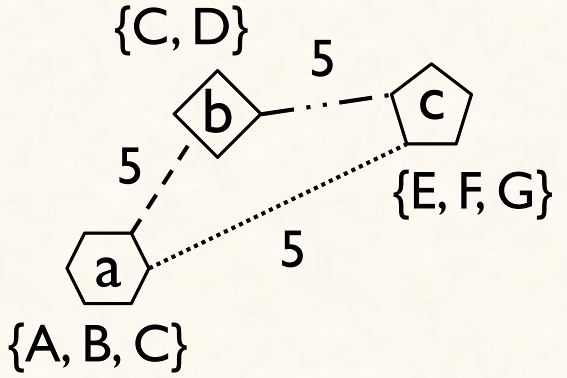
*Ye, Zilong, *et al.* "Survivable Virtual Infrastructure Mapping With Dedicated Protection in Transport Software Defined Networks." *Journal of Optical Communications and Networking* 7(2): A183-A189, 2015.

OPT-DRONE:

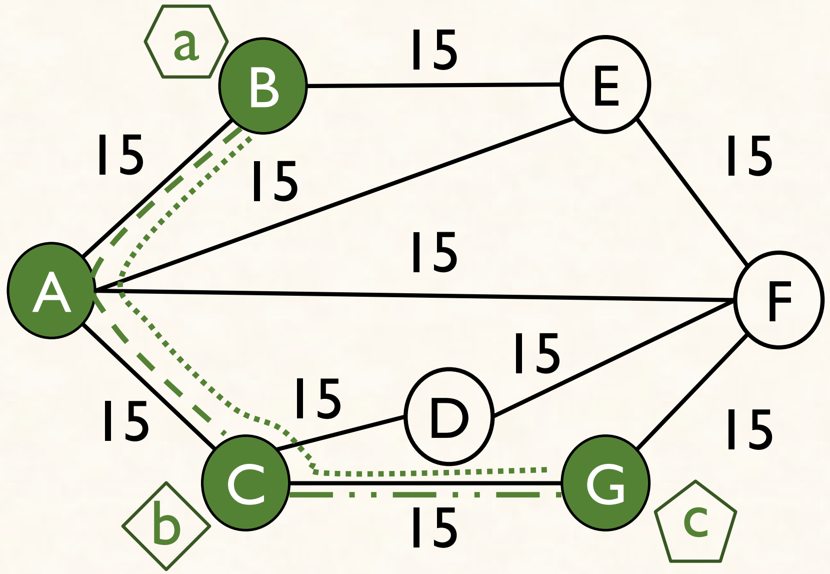
Integer Linear Program (ILP) model for optimal solution to 1 + 1-ProViNE that minimizes physical bandwidth allocation cost


Heuristic Design
Take 1: Sequential Embedding

Embed the VN first
(primary embedding)



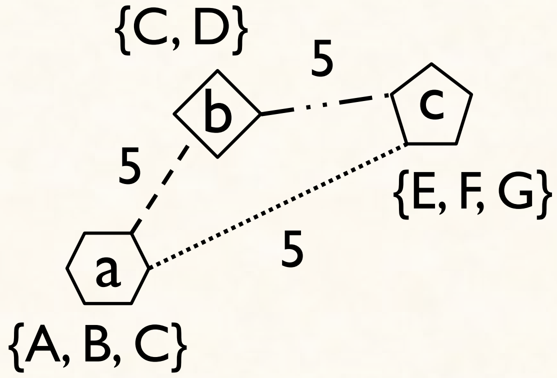
Virtual Network (G')



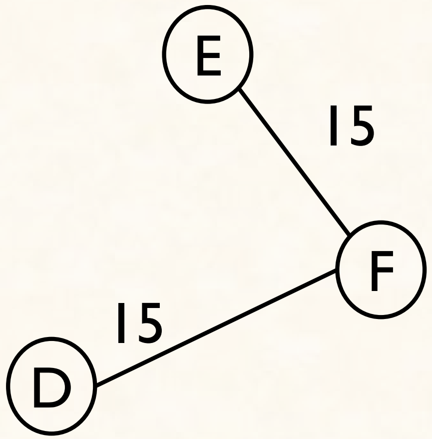
Physical Network (G)

Embed the VN first
(primary embedding)

Remove physical resources
in primary embedding



Virtual Network (G')

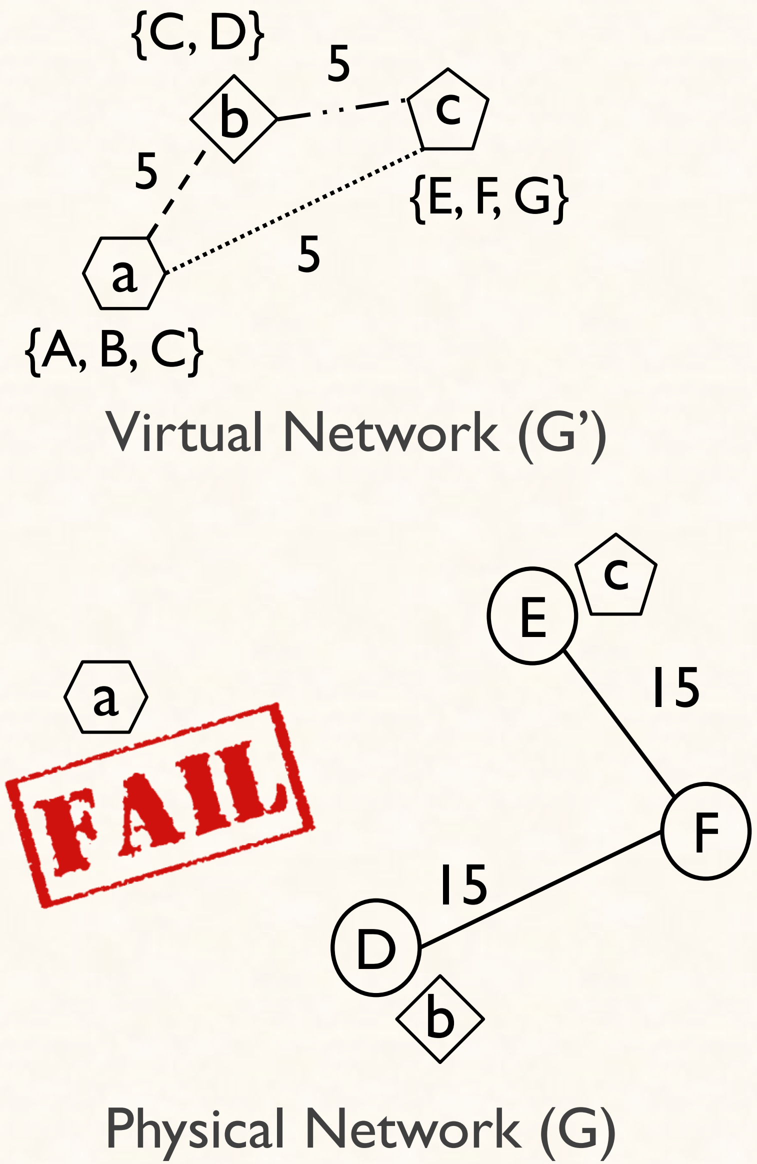


Physical Network (G)

Embed the VN first
(primary embedding)

Remove physical resources in primary embedding

Embed the VN again
(backup embedding)



Why Sequential Embedding can Fail?

Physical Node and Link selection for primary can make backup embedding infeasible

Why Sequential Embedding can Fail?

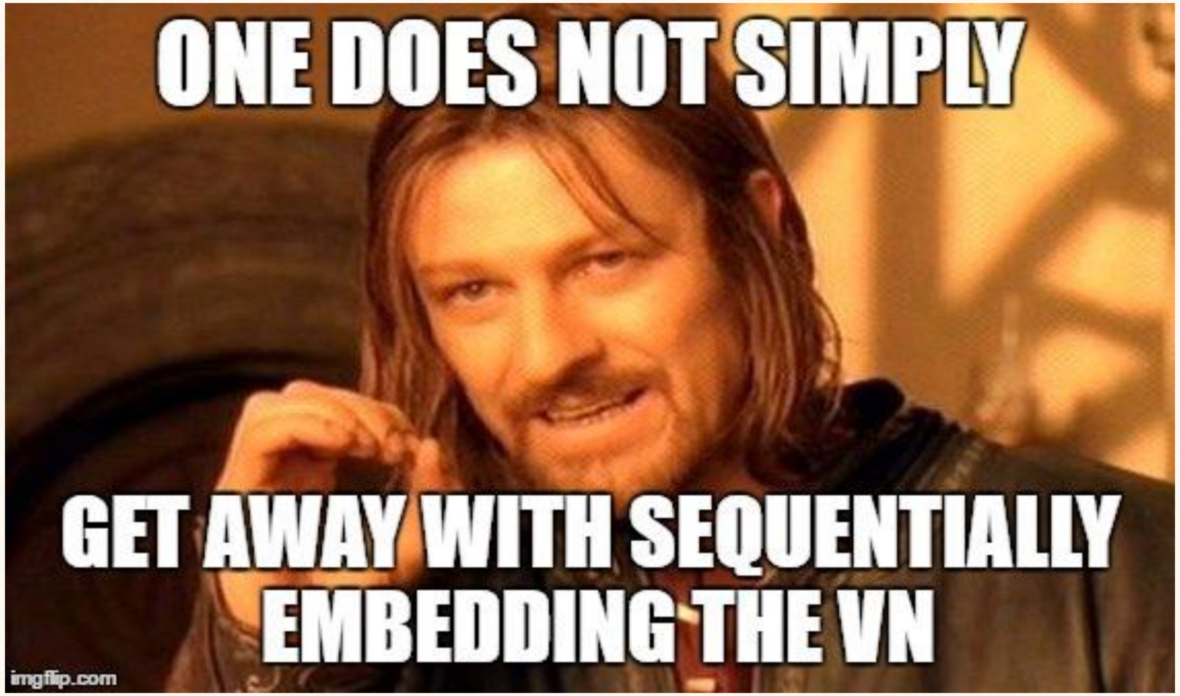
Physical Node and Link selection for primary can make backup embedding infeasible

Not Considered by Sequential Embedding!

Take Away

Embed primary and backup at the same time, not sequentially

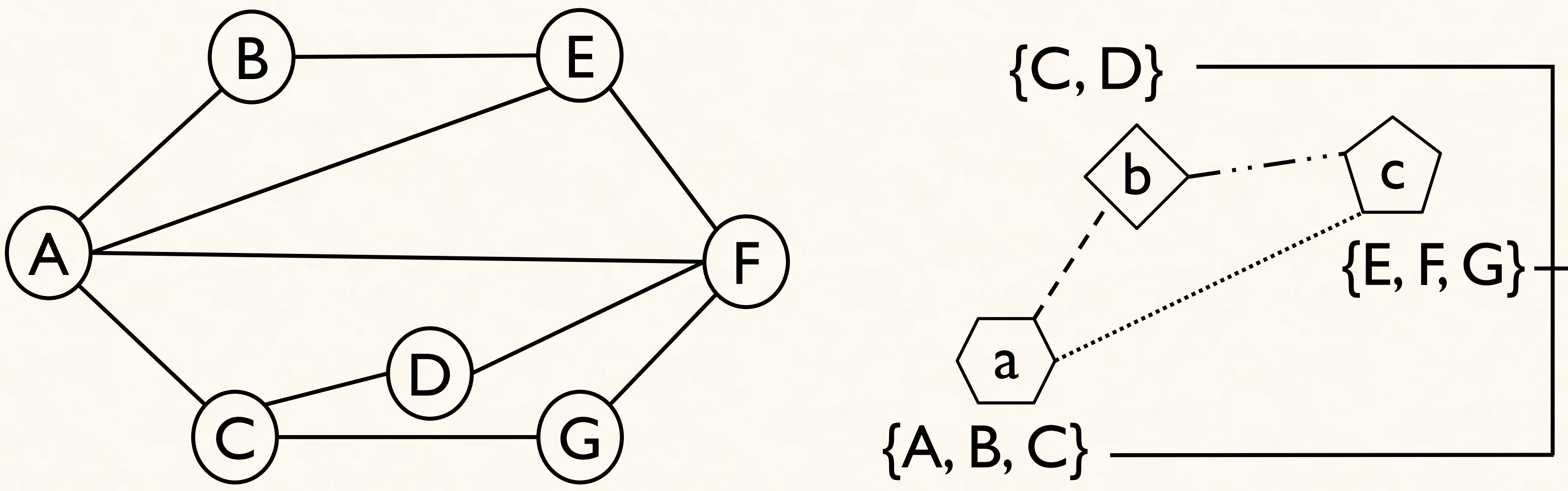
Consider infeasibility of backup embedding while doing the primary




Heuristic Design
Take n: FAST-DRONE

FAST-DRONE: Solution Approach

Reformulate the problem as a special case of Graph Partitioning



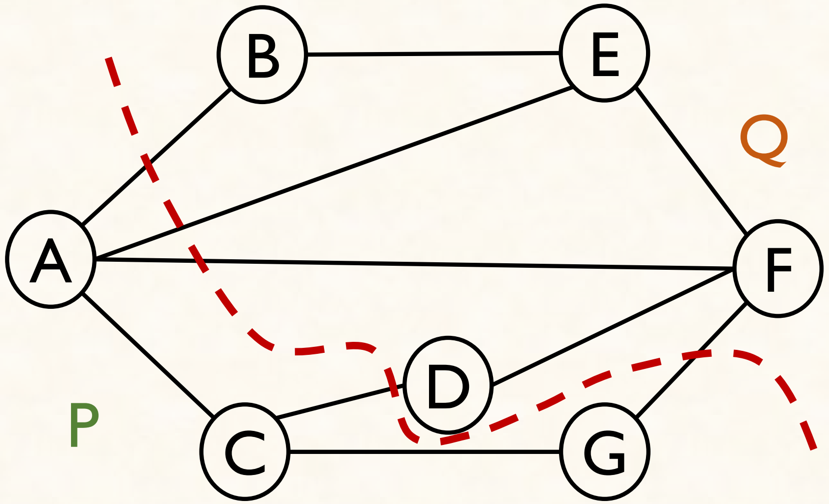
A Set of location constraints
$L = \{L_{u'} \mid L_{u'} \subseteq V, \forall\, u' \in V'\}$

Physical Network Graph
$G = (V, E)$

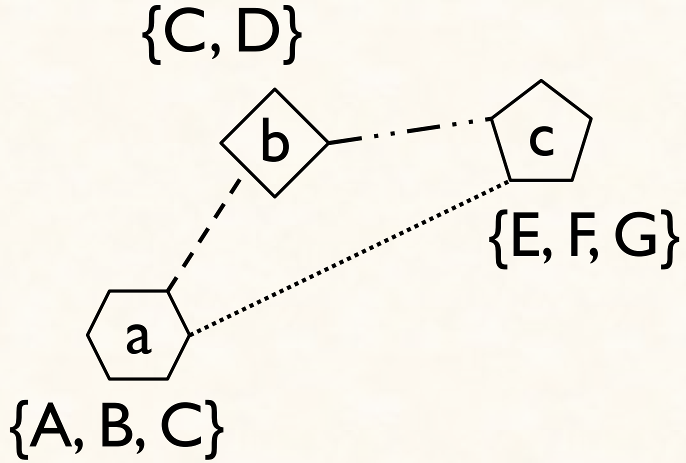
Virtual Network Graph
$G' = (V', E')$

FAST-DRONE: Solution Approach

Partition G into two node-disjoint partitions P and Q for the primary and the backup embedding



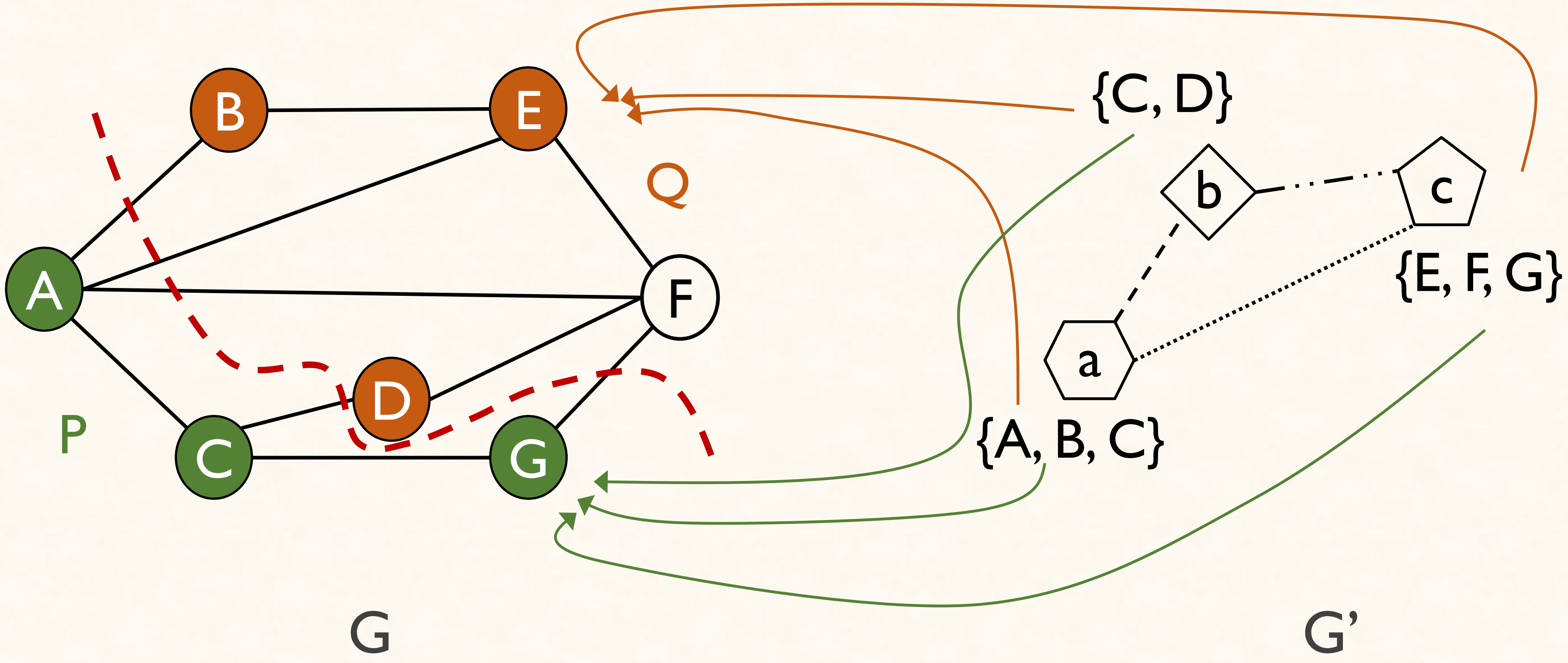
G



G’

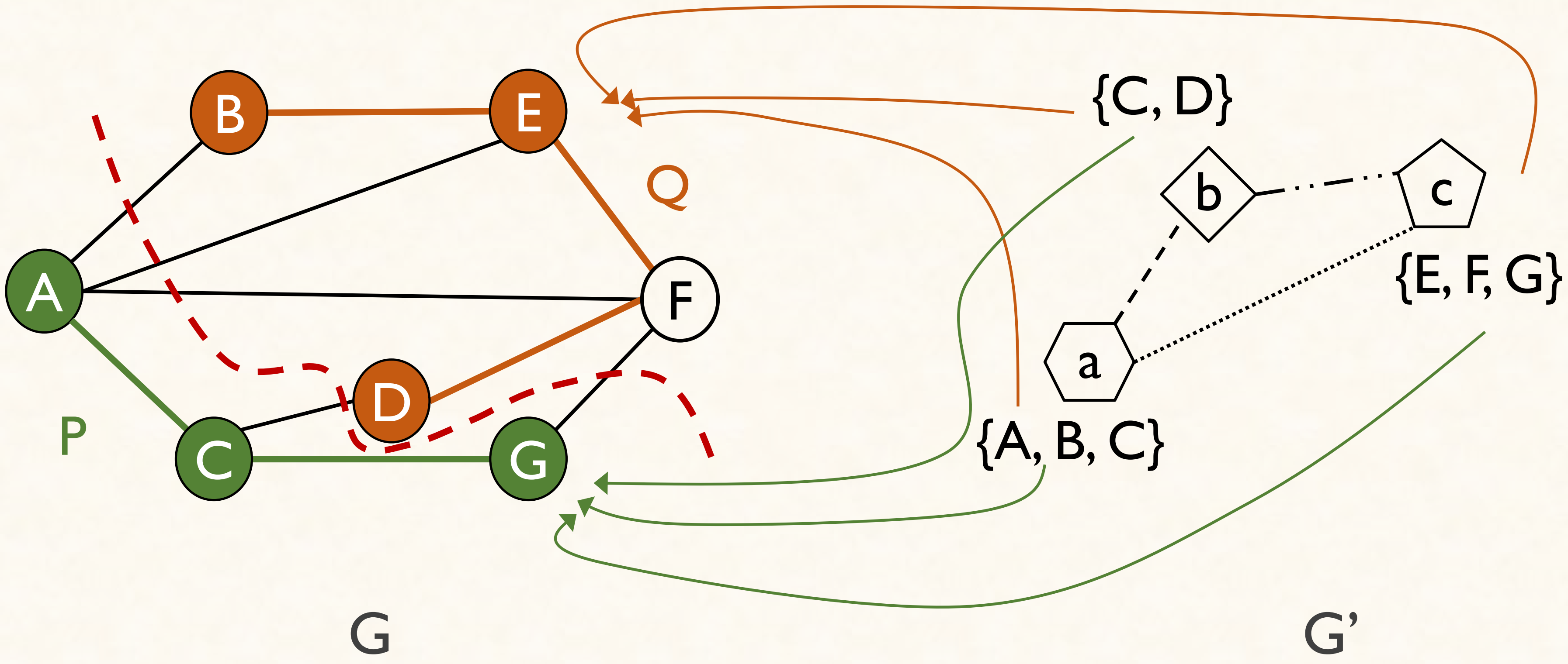
FAST-DRONE: Solution Approach

P and Q each satisfies at least one location constraint from each set $L_{u'}$



FAST-DRONE: Solution Approach

The nodes satisfying the location constraints of $L_{u'}$ in P and Q are connected



What does the restructuring say?

1+1-ProViNE is at least as hard as jointly solving *Balanced Graph Partitioning* and *Unsplittable Multi-commodity flow with unknown sources and destinations*

FAST-DRONE: 3-phase algorithm

Intuition from restructuring: Partition the Physical Network in some way and embed the VN & backup in parallel

FAST-DRONE: 3-phase algorithm

Intuition from restructuring: Partition the Physical Network in some way and embed the VN & backup in parallel

Phase-I: Compute primary and backup embedding of virtual nodes

FAST-DRONE: 3-phase algorithm

Intuition from restructuring: Partition the Physical Network in some way and embed the VN & backup in parallel

Phase-I: Compute primary and backup embedding of virtual nodes

Phase-II: Partition the physical network based on node embedding

FAST-DRONE: 3-phase algorithm

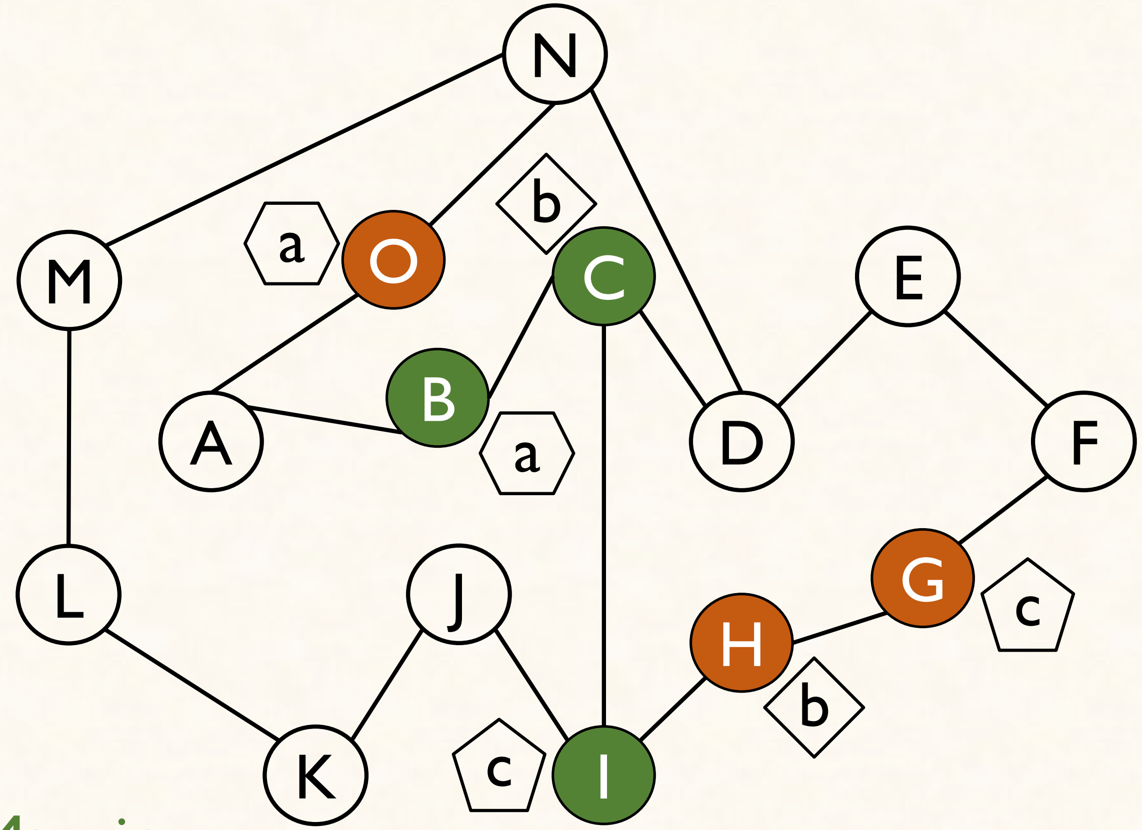
Intuition from restructuring: Partition the Physical Network in some way and embed the VN & backup in parallel

Phase-I: Compute primary and backup embedding of virtual nodes

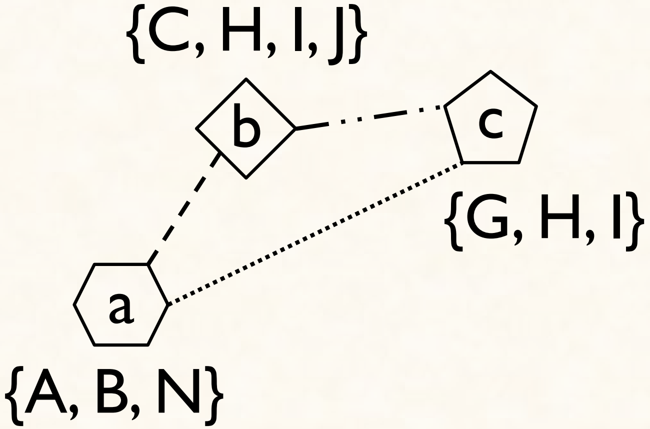
Phase-II: Partition the physical network based on node embedding

Phase-III: Embed virtual links inside primary and backup partitions

Phase – I: Node Mapping Phase



Physical Network - G

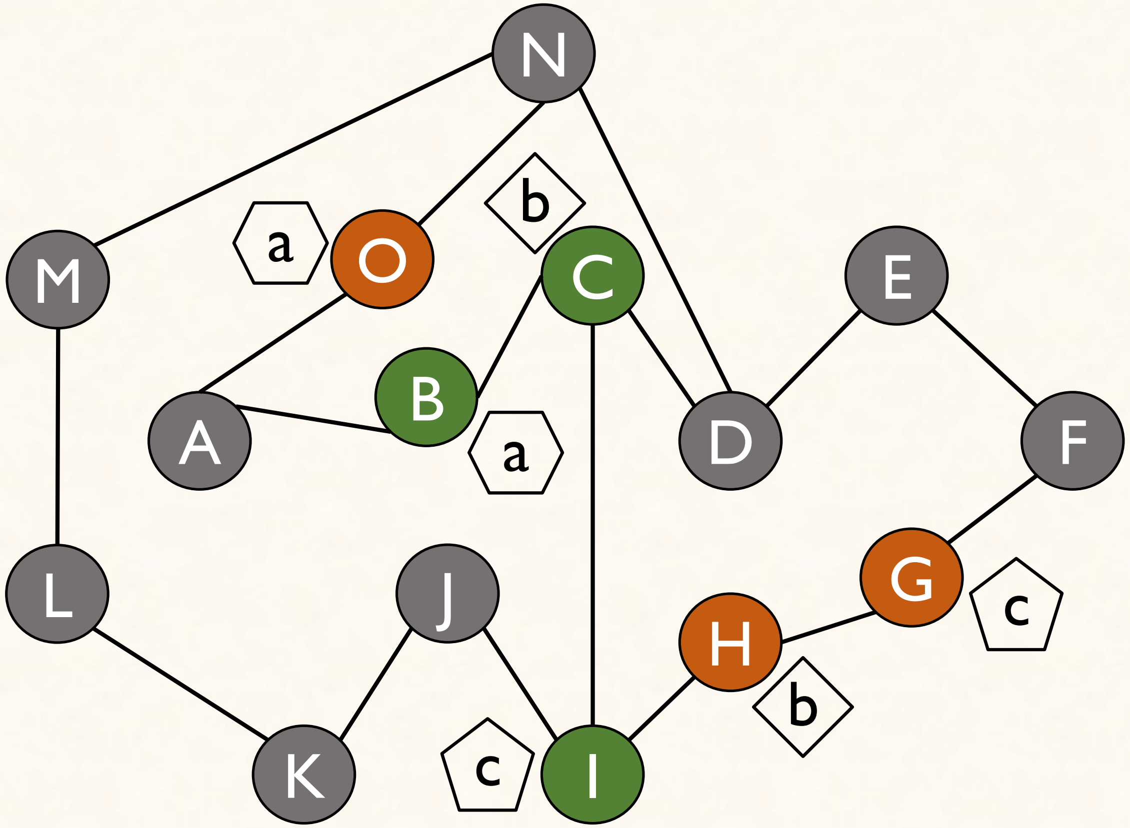


Virtual Network - G'

Output:
Primary Mapping
Backup Mapping

Phase – II: Partitioning Phase

Input:
Seed Primary Partition
Seed Backup Partition
Un-partitioned Nodes

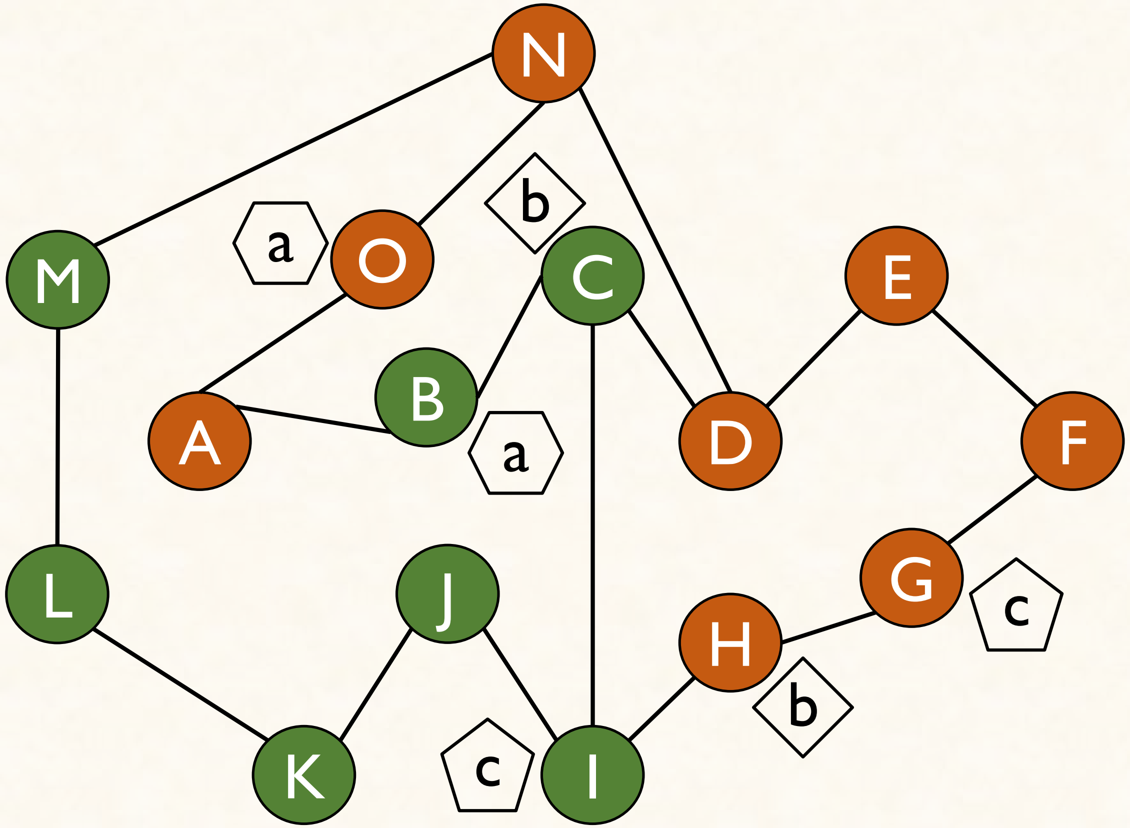


Physical Network - G

Phase – II: Partitioning Phase

Grow each seed partition to include all physical nodes.

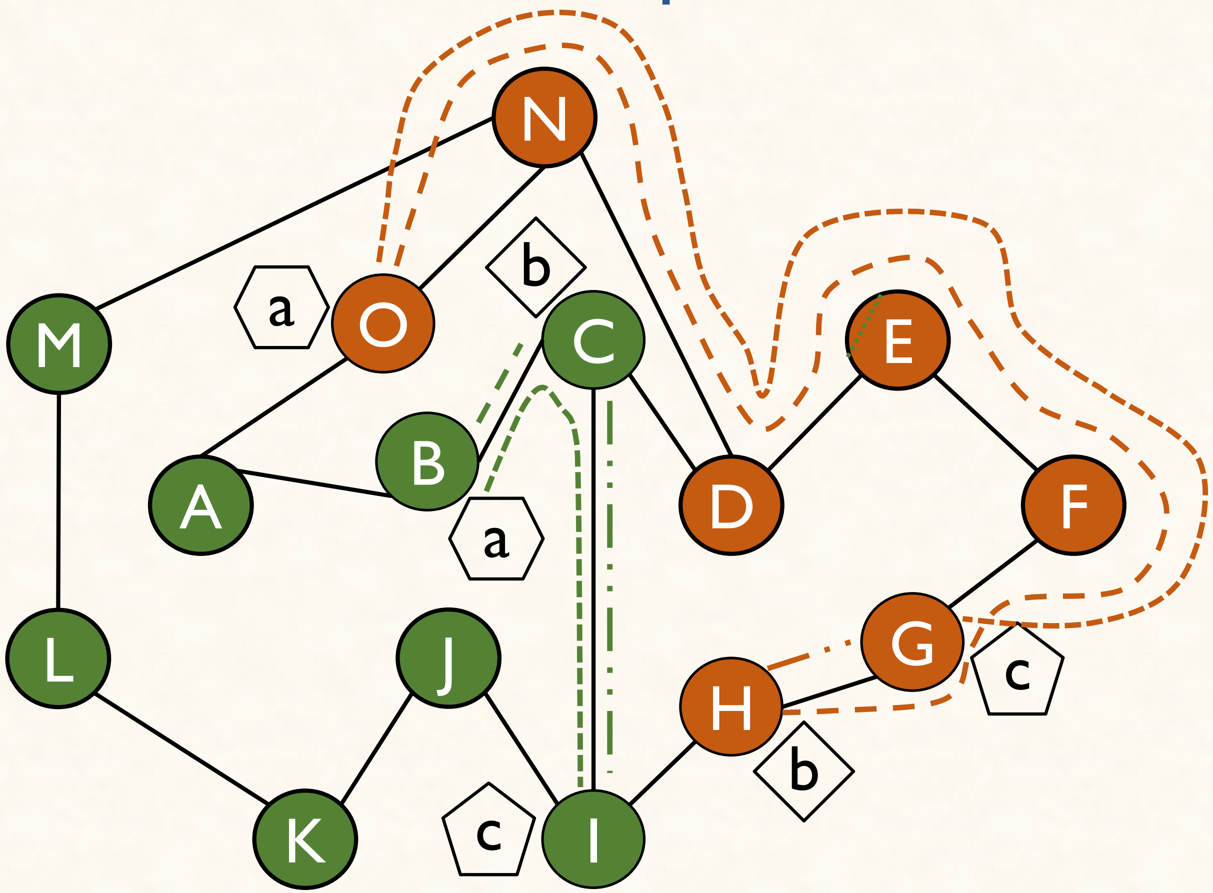
<u>Output:</u>
Primary Partition
Backup Partition



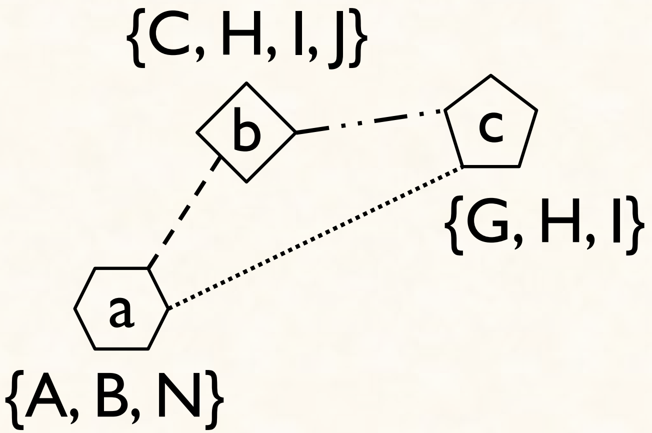
Physical Network - G

Phase – III: Link Mapping Phase

Map virtual links on shortest path between their mapped endpoints



Physical Network - G



Virtual Network - G'

Evaluation: Setup

- FAST-DRONE compared with OPT-DRONE and PAR*
- Physical Network
  - 50 – 150 node synthetic topology
  - Mean degree between 2.4 – 4.4
- Virtual Network with <= 16 nodes

* Ye, Zilong, *et al.* "Survivable Virtual Infrastructure Mapping With Dedicated Protection in Transport Software Defined Networks." *Journal of Optical Communications and Networking* 7(2): A183-A189, 2015.

FAST-DRONE Performance Highlights

Within 15% of optimal on avg.

17.5% better than PAR on avg.

2-3 Orders of magnitude faster than Optimal

Summary

Two solutions to 1+1-ProViNE: OPT-DRONE, FAST-DRONE

FAST-DRONE Outperforms stat-of-the-art

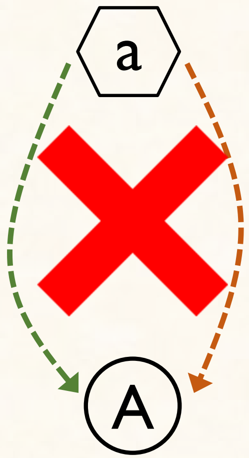
FAST-DRONE performs within ~15% of the Optimal

Questions?

+ 1 Backup

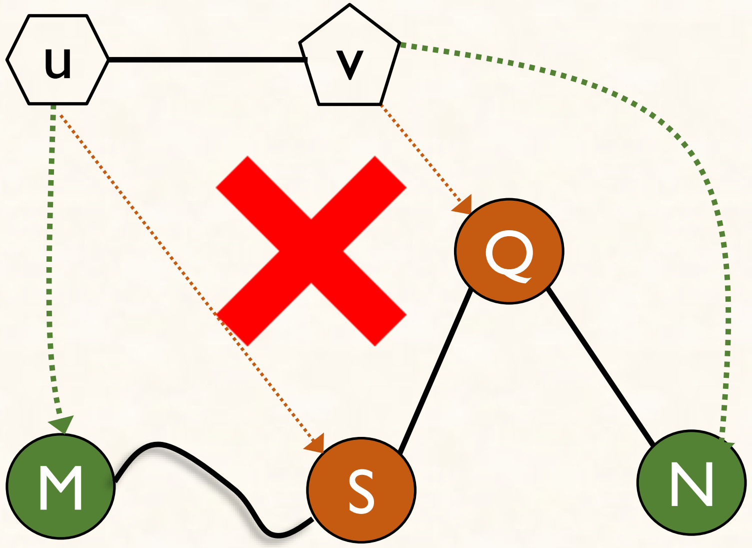
OPT-DRONE: Key Constraints

A virtual node's primary embedding is disjoint from the set of physical nodes present in the virtual network's backup embedding and vice versa.

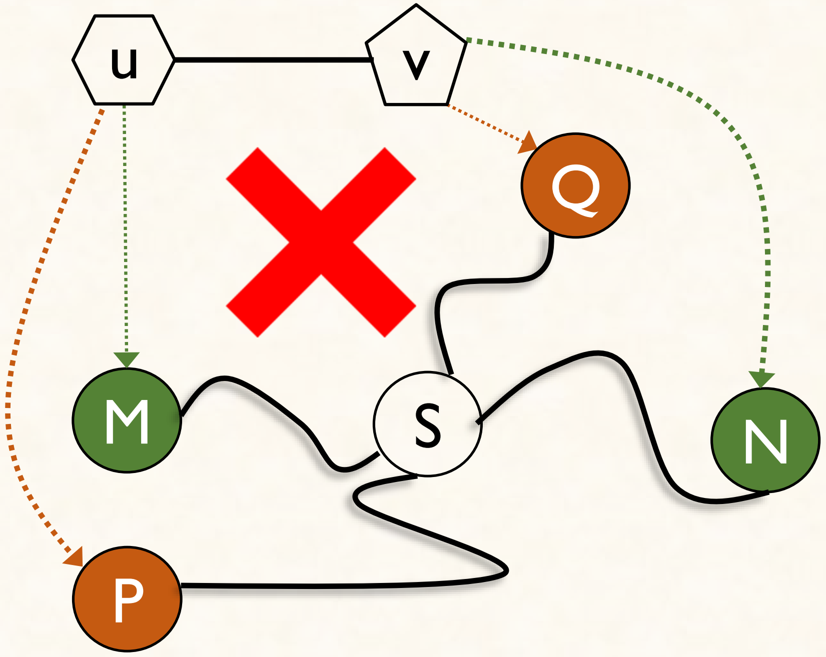


OPT-DRONE: Key Constraints

A virtual link's primary embedding is link and node disjoint from the set of physical links present in the virtual network's backup embedding and vice versa.



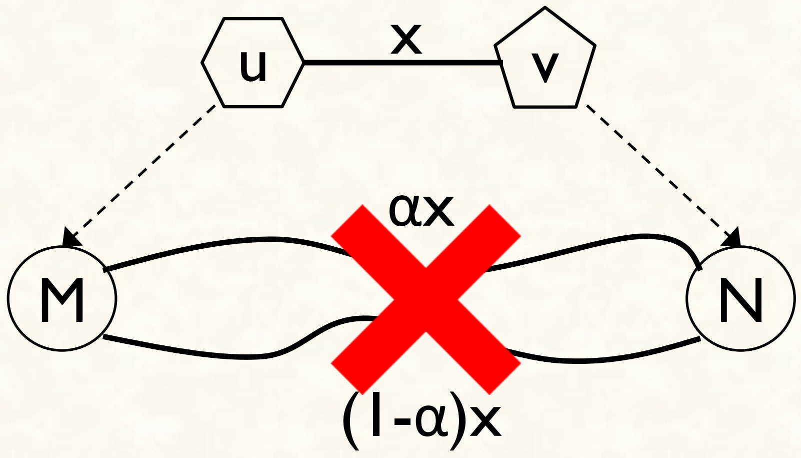
No Link Sharing



No Node Sharing

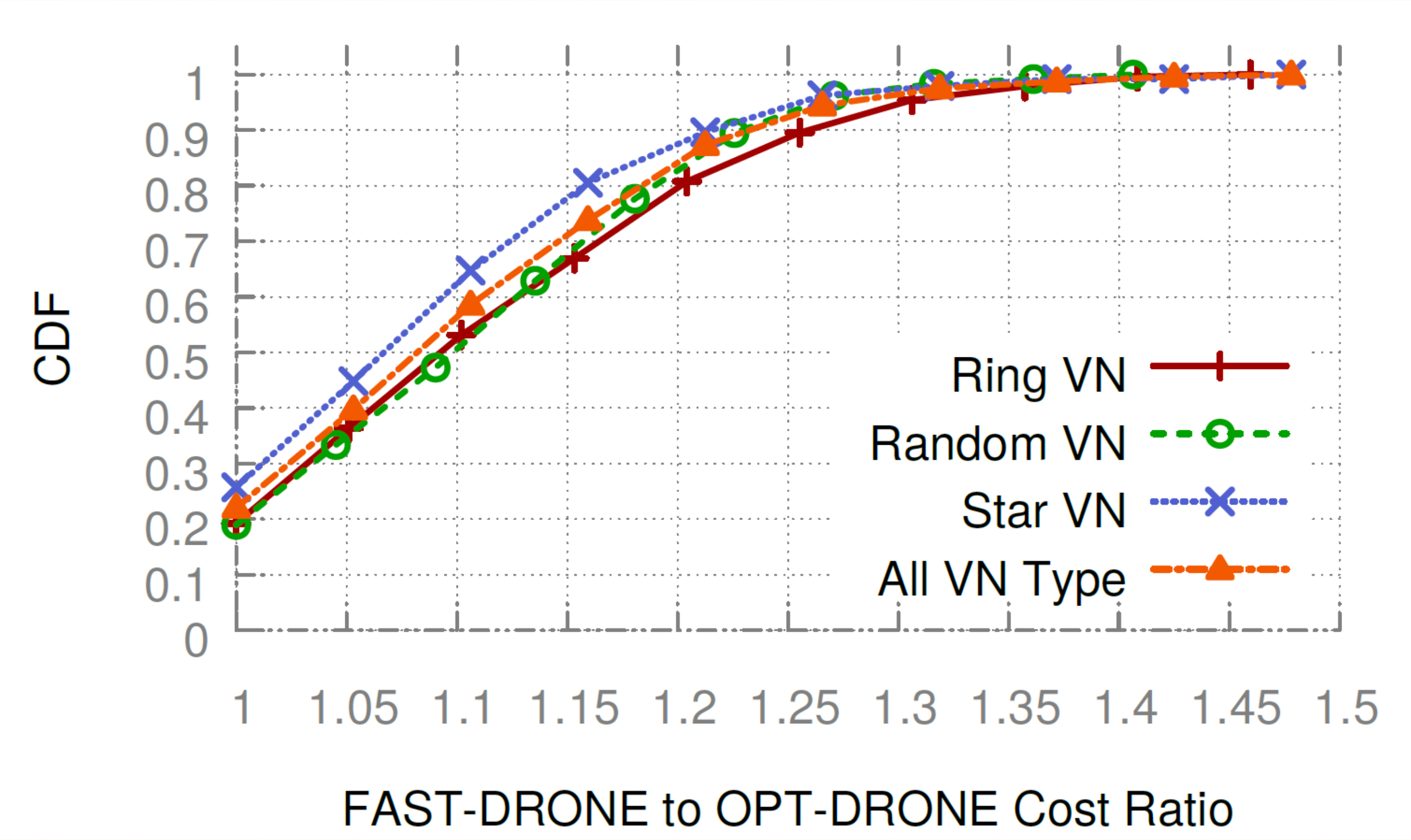
OPT-DRONE: Key Constraints

A virtual link's demand cannot be satisfied using multiple physical paths.

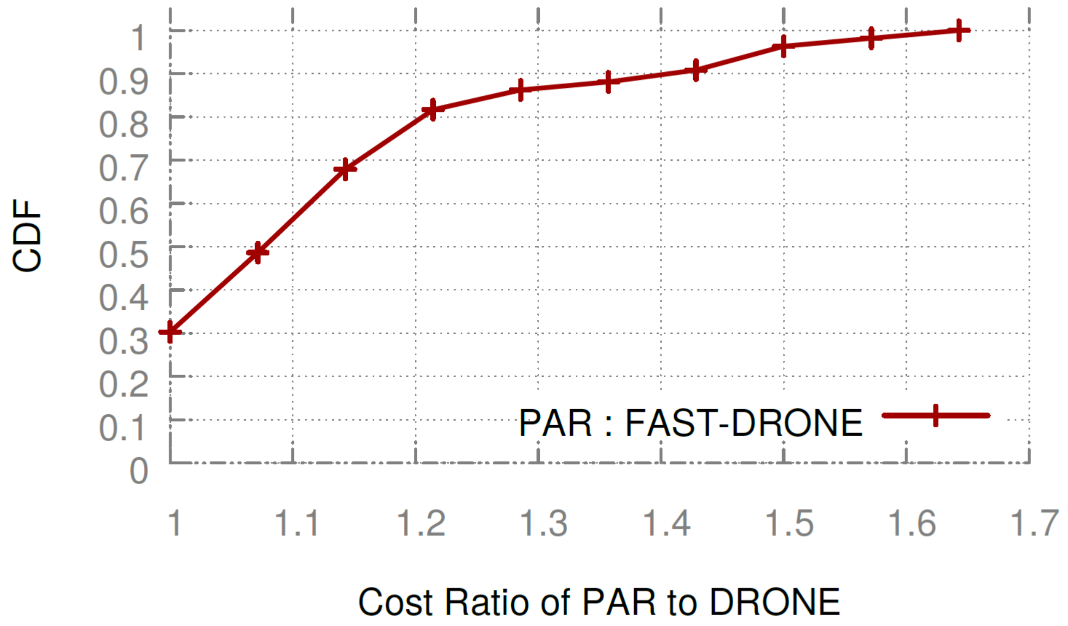


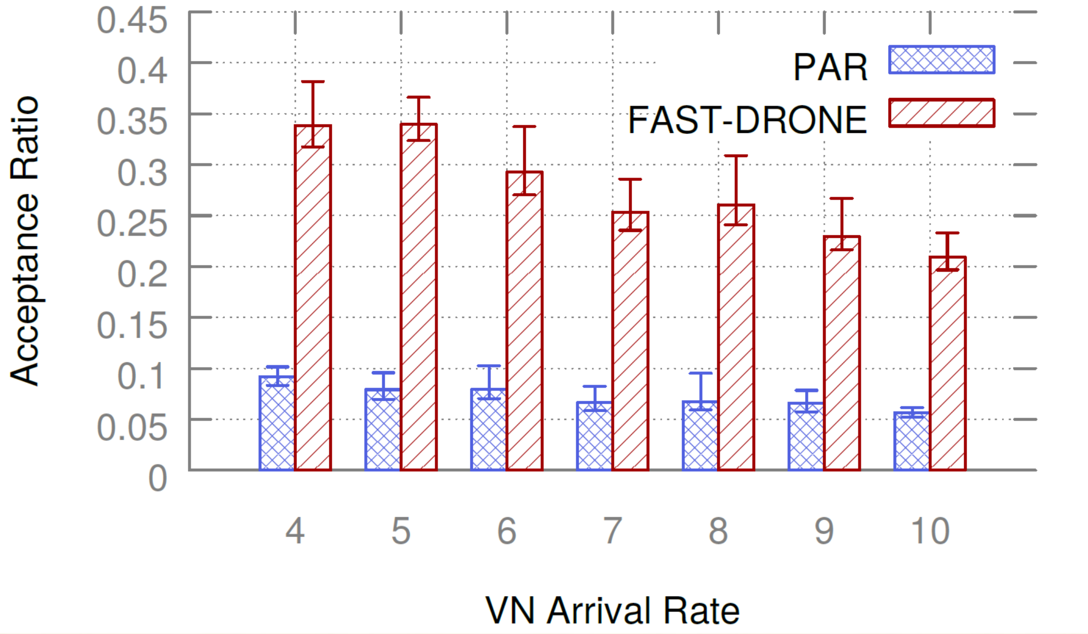
At least as hard as *Unsplittable Multi-Commodity Flow with Unknown Source & Destination* (NP-Hard)

FAST-DRONE vs OPT-DRONE



FAST-DRONE vs PAR





Resource Efficiency

Acceptance Rate
(From Journal Submission)