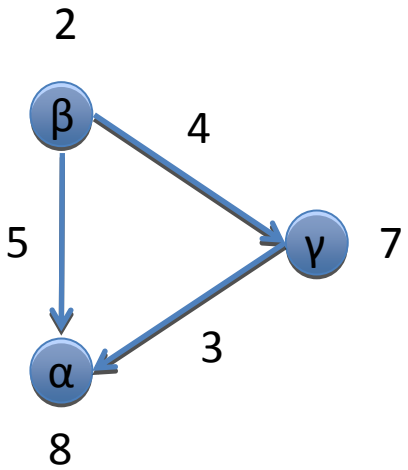
Virtual Network Mapping based on Subgraph Isomorphism Detection

Jens Lischka, Holger Karl

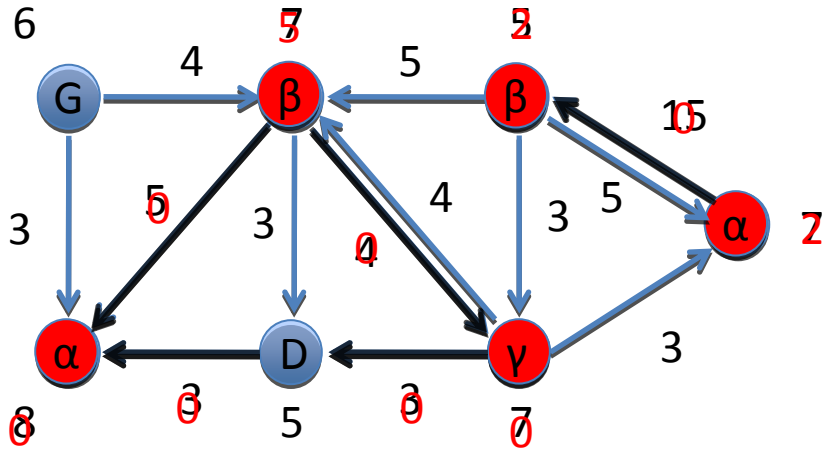
Paderborn University

VNM Problem

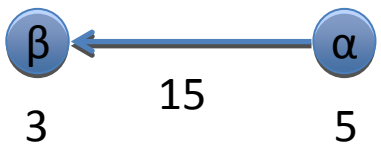
VNR 1(t_0, 10)



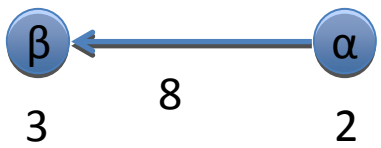
PN



VNR 2(t_5,3)



VNR 3(t_6,3)

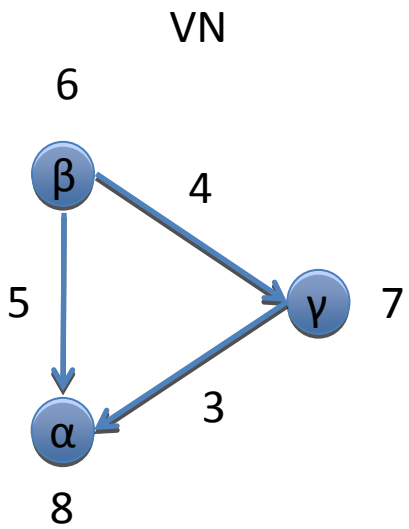


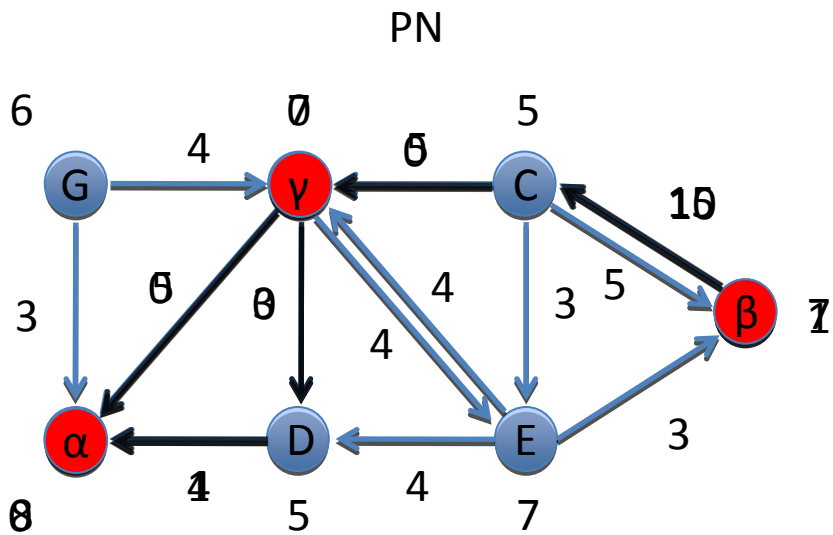
Overview

- 2stage VNM algorithm
- Subgraph Isomorphism Detection based VNM
- Experimental results

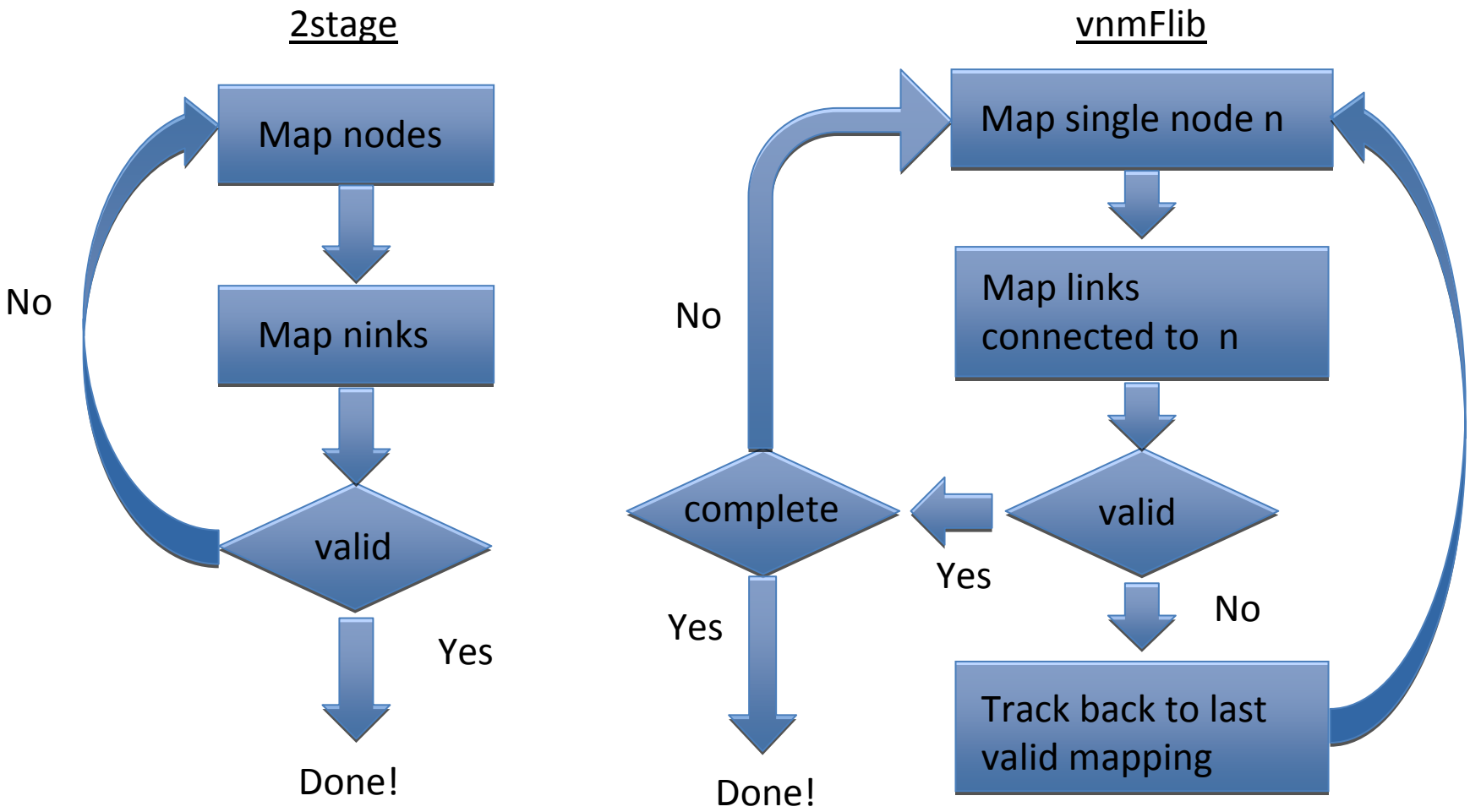
2stage Algorithm

1. First stage: find suitable mapping nodes
2. Second stage: find a link mapping (k-shortest paths, multi commodity flow)
3. No paths for virtual links β->γ!
4. Problem: first stage does not take connectivity of VNs into account





2stage vs. vnmFlib

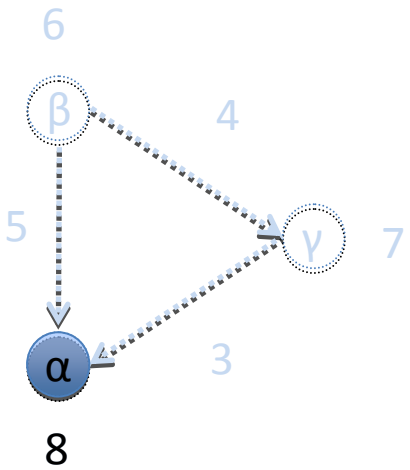


Example: vnmFlib

1. Compute set of candidates C.
2. Compute a set of mapping candidates M.
3. Add α to the subgraph and map it onto A.
4. Map all links connecting α with the subgraph onto the PN
5. Check validity.

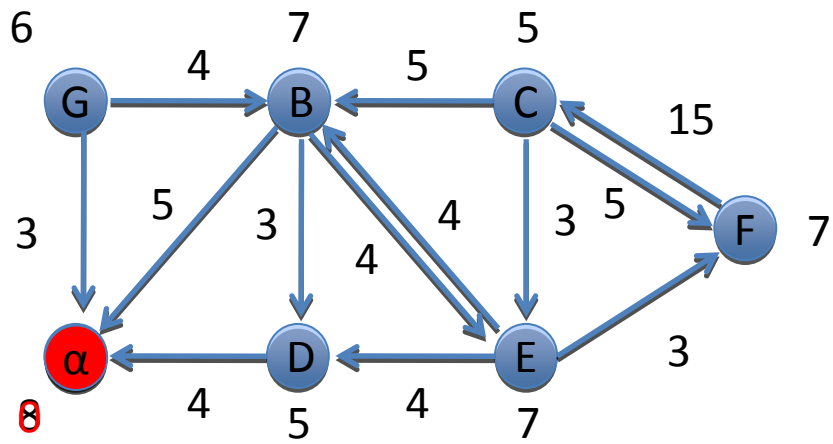
Subgraph

C={α,γ,β}



Mapping

M={A}

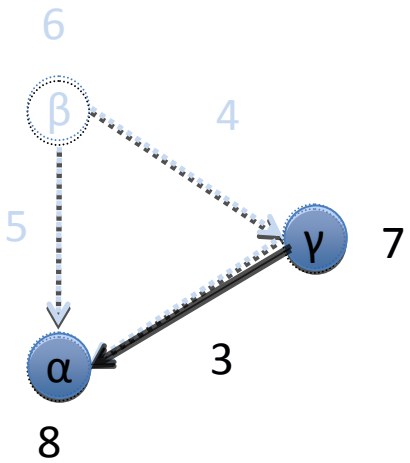


Example: vnmFlib

1. Compute C and M.
2. Add γ to the subgraph and map it onto B.
3. Map all links connecting γ with the subgraph onto the PN.
4. Check validity.

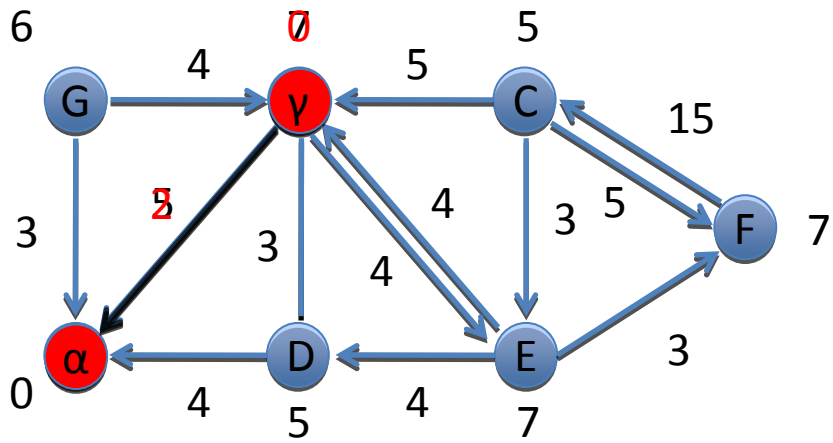
Subgraph

C={γ,β}



Mapping

M={B,E,F}

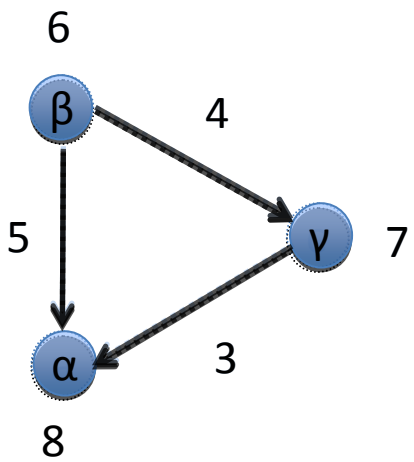


Example: vnmFlib

1. Compute C and M.
2. Add β to the subgraph and map it onto G.
3. Map all links connecting β with the subgraph onto the PN.
4. Check validity.

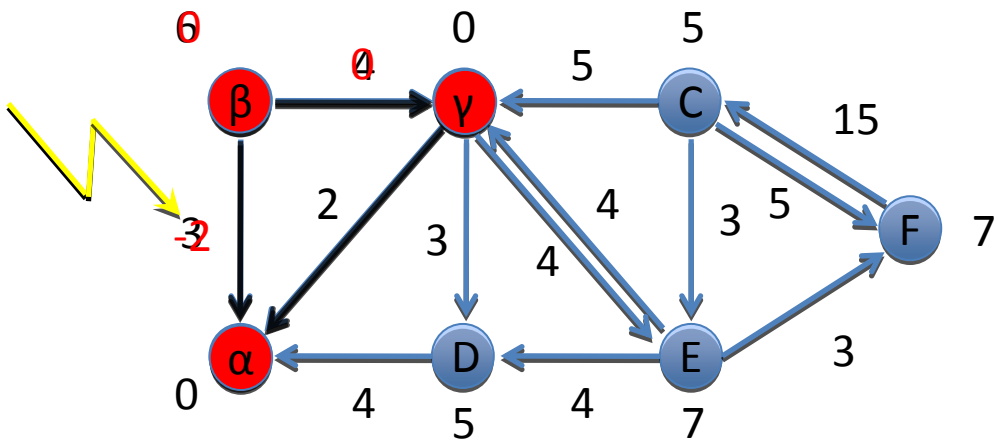
Subgraph

C={β}



Mapping

M={G,E,F}

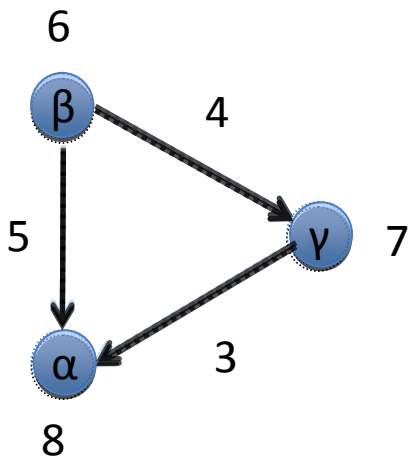


Example: vnmFlib

1. Choose next node E of M.
2. Map β onto E.
3. Map all links connecting β with the subgraph onto the PN.
4. Check validity.

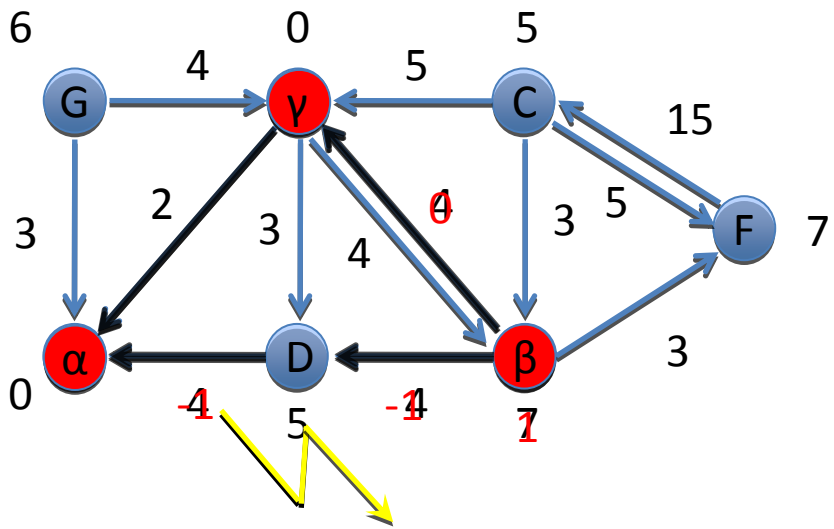
Subgraph

C={b}



Mapping

M={G,E,F}

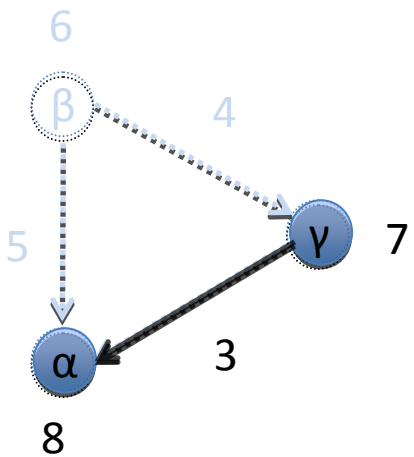


Example: vnmFlib

1. Track back to the last valid mapping solution.
2. Choose next node E.
3. Map γ onto E.
4. Map all links connecting γ with the subgraph onto the PN.
5. Check validity.

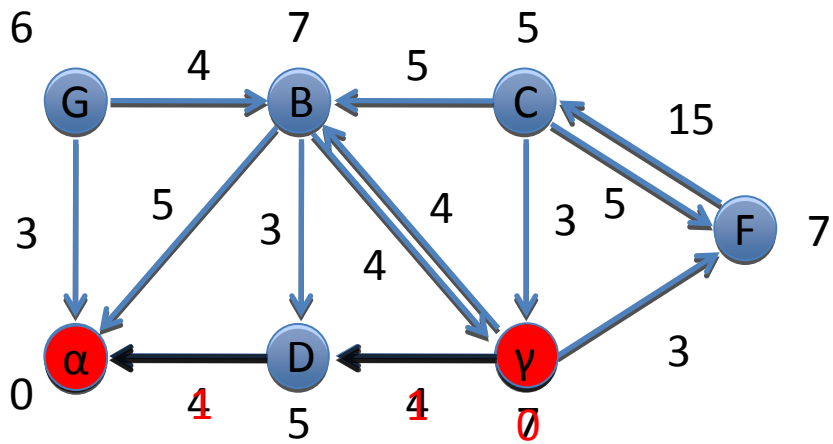
Subgraph

C={γ,β}



Mapping

M={B,E}

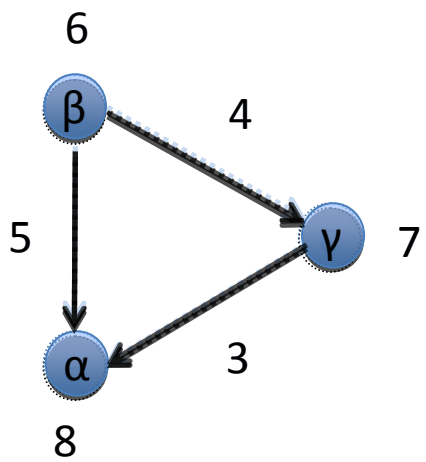


Example: vnmFlib

1. Compute C and M.
2. Add β to the subgraph and map it onto B.
3. Map all links connecting β with the subgraph onto the PN.
4. Check validity.

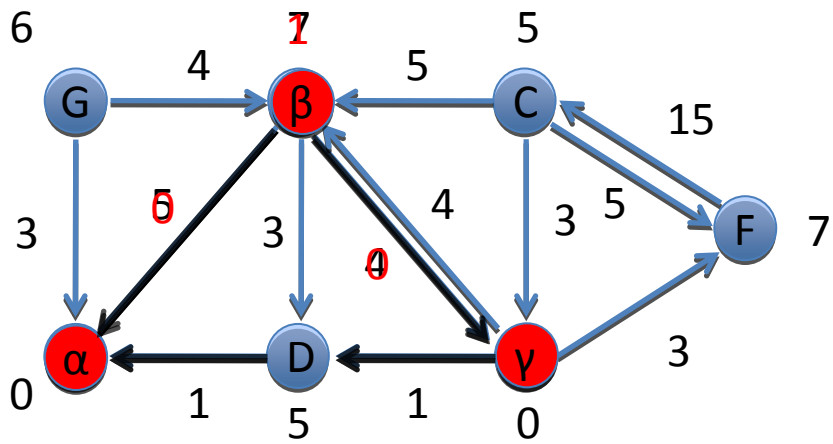
Subgraph

C={β}



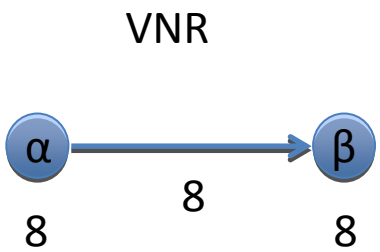
Mapping

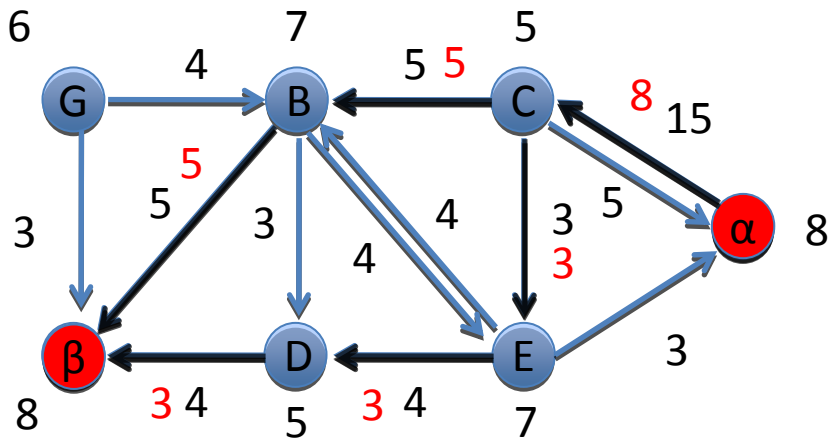
M={B,F,G}



Path Splitting

- Split up path into multiple paths





Experimental Results

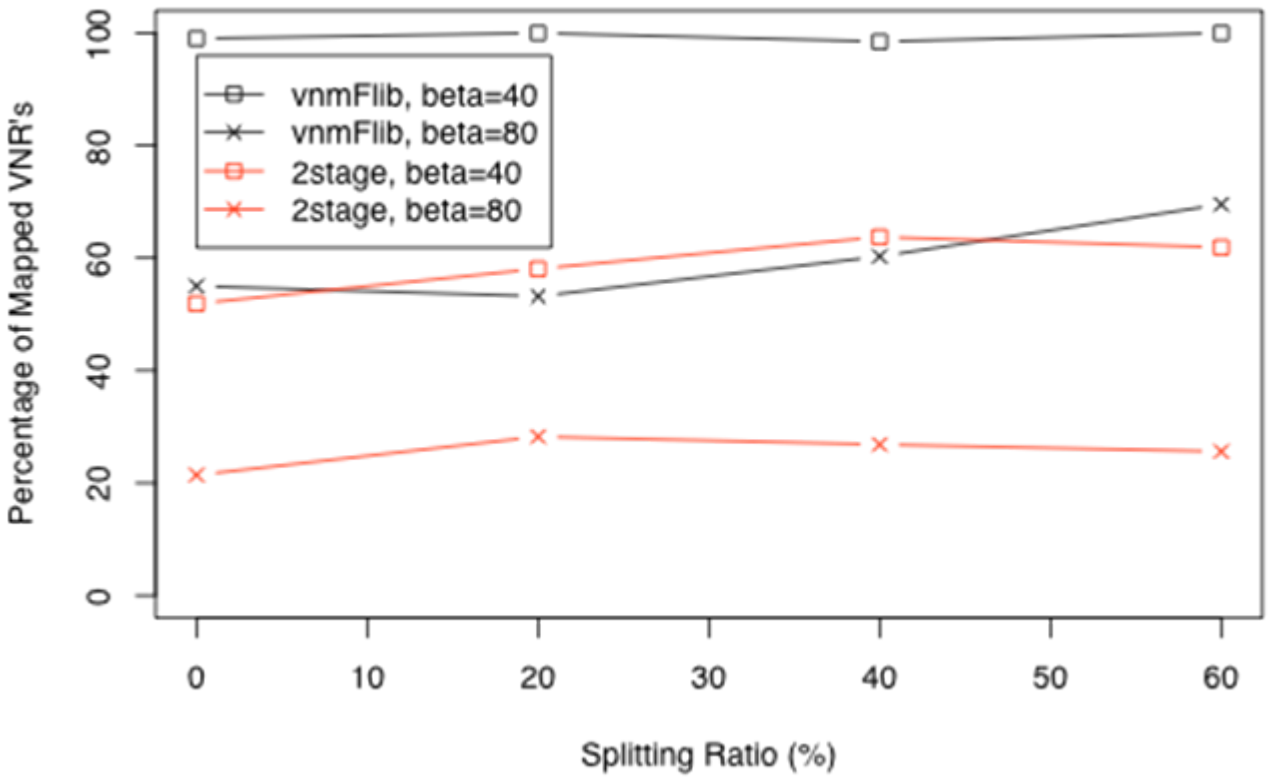
- Network setup similar to previous work[1] with GT-ITM tool:
  - PN: 100 nodes and 500 links

    CPU at the nodes, Bandwidth at the links follow uniform distribution from 0-100 units
  - VNs: 20-40 nodes, each pair of nodes connected with probability 0.5

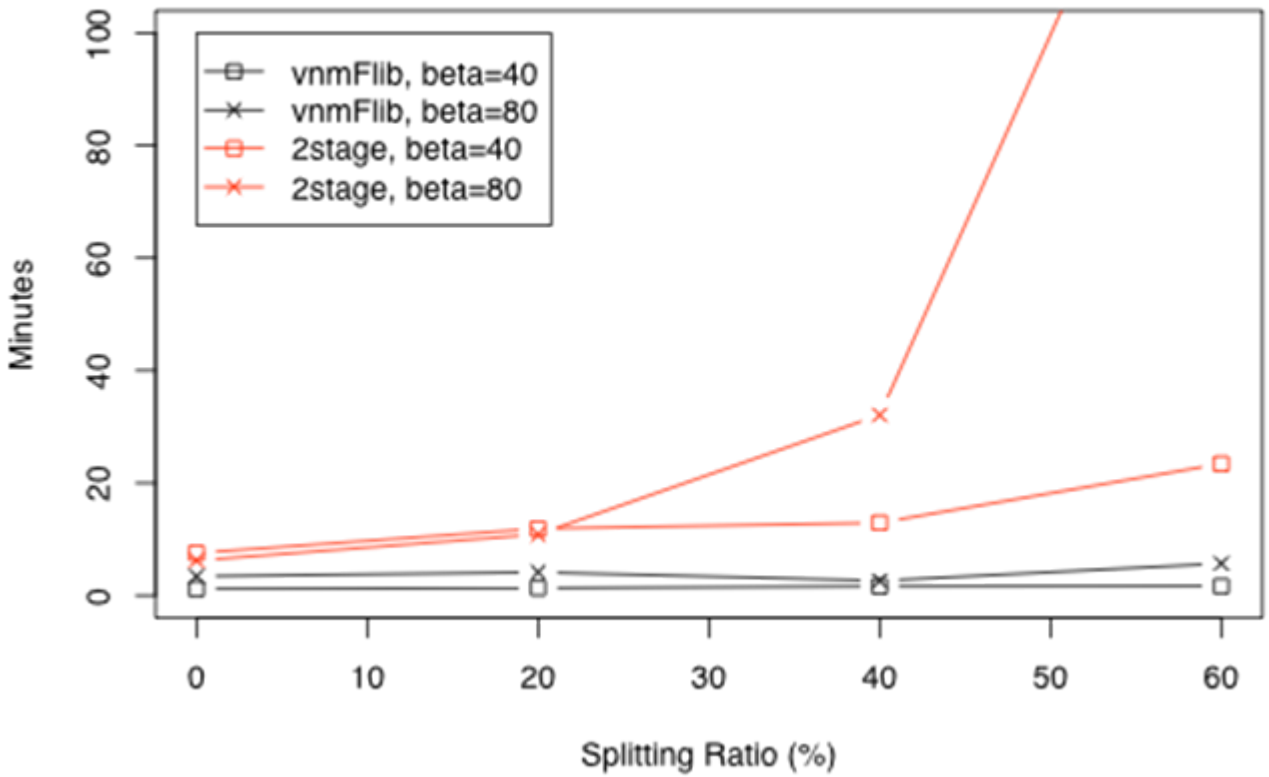
    CPU and Bandwidth follow a uniform distribution from 0 to beta units.
- Compared our algorithm with the two stage VN Mapper of [1].

[1]Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. SIGCOMM Comput. Commun. Rev., 38(2):17-29, 2008.
Source code available: http://www.princeton.edu/~minlanyu/embed.tar.gz

Experimental results



Experimental Results



Summary

- Introduced new VNM method based on SID
- SID based VNM performs better than the 2stage approach
  - Especially for higher beta values and bigger networks
- Currently we are implementing the mapper on the PlanetLabTestbed infrastructure as part of the OneLab2 project.

Thank You

Questions?

VNM Algorithms

- 2stage:
  - Rethinking Virtual Network Embedding: Support for Path Splitting and Migration. SIGCOMM, 2008.
  - Algorithms for Assigning Substrate Network Resources to Virtual Network Components, INFOCOMM, 2006.
  - A Multi-Commodity Flow Based Approach to Virtual Network Resource Allocation. GLOBECOMM, 2003.
- Simulated Annealing:
  - A Solver for the Network Testbed Mapping Problem. Computer Communications Review 33(2), 2003.
- Mixed Integer Quadratic Program
  - Efficient Mapping of Virtual Networks onto a shared Substrate. Technical Report, Washington University.

VNM Algorithms

- Virtual Network Embedding with Coordinated Node and Link Mapping.
  - In Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOMM), April 2009.

SID based VNM

- Idea: Map Nodes and Links alternately based on vFlibSubgraph Isomorphism Detection algorithm.
- Build a subgraph S of VN by successively adding nodes of VN to S and map S onto PN until S fully covers VN.
- Difference to vFlib:
  - Allow mapping of virtual links onto paths
  - Check capacity constraints