Virtual Network Mapping based on Subgraph Isomorphism Detection

Jens Lischka, Holger Karl
Paderborn University
VNM Problem

VNR 1(t₀, 10)

VNR 2(t₅, 3)

VNR 3(t₆, 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 \( \beta \rightarrow \gamma \)!
4. Problem: first stage does not take connectivity of VNs into account
2stage vs. vnmFlib

2stage

Map nodes

Map ninks

valid

Yes

Done!

No

Map ninks

2stage

vnmFlib

Map single node n

Map links connected to n

valid

Yes

Done!

No

complete

Yes

Yes

Track back to last valid mapping

No

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

Subgraph

$C = \{\alpha, \gamma, \beta\}$

Mapping

$M = \{A\}$

Example: vnmFlib
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

\[ C = \{\gamma, \beta\} \]

Mapping

\[ M = \{B, E, F\} \]
1. Compute $C$ and $M$.
2. Add $\beta$ to the subgraph and map it onto $G$.
3. Map all links connecting $\beta$ with the subgraph onto the PN.
4. Check validity.

Subgraph

$C=\{\beta\}$

Mapping

$M=\{G,E,F\}$
1. Choose next node E of M.
2. Map \( \beta \) onto E.
3. Map all links connecting \( \beta \) with the subgraph onto the PN.
4. Check validity.

Subgraph

\[ C = \{ b \} \]

Mapping

\[ M = \{ G, E, F \} \]
1. Track back to the last valid mapping solution.
2. Choose next node E.
3. Map $\gamma$ onto E.
4. Map all links connecting $\gamma$ with the subgraph onto the PN.
5. Check validity.

$$C = \{ \gamma, \beta \}$$

$$M = \{ B, E \}$$
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].

Source code available: http://www.princeton.edu/~minlanyu/embed.tar.gz
Experimental results

![Graph of Experimental Results]

- vnmFlib, beta=40
- vnmFlib, beta=80
- 2stage, beta=40
- 2stage, beta=80

Percentage of Mapped VNR's vs. Splitting Ratio (%)
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:

• Simulated Annealing:

• Mixed Integer Quadratic Program
• Virtual Network Embedding with Coordinated Node and Link Mapping.
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