Specificity vs. Flexibility:
On the Embedding Cost of a Virtual Network

Arne Ludwig,
Stefan Schmid, Anja Feldmann

Telekom Innovation Laboratories & TU Berlin, Germany
{arne,stefan,anja}@net.t-labs.tu-berlin.de

September 14, 2013
Motivation
Motivation
Motivation
Motivation
Motivation
Motivation

▶ Location: Italy

▶ Location: Spain
Motivation

- Location: Italy
- Location: Spain
Motivation

- Location: Italy
- Location: Spain
Standard Problem
How can flexibility be quantified?
Focus on node properties

- Different properties:
  - Location
  - Virtualization technology
  - Operating system
  - etc.

- Substrate nodes have exactly one type per property
  - VNet node requests can specify multiple types

- All properties combined form a configuration
  - Example: \{Italy, Xen, RedHat 7.3\}
Specificity - Definition

Specificity \( \sigma = \) percentage of lost alternatives

\[ \Rightarrow \sigma = \frac{\text{forbidden configurations}}{\text{all configurations} - 1} \]

- \( \sigma = 0 \): free choice of nodes
- \( \sigma = 1 \): only nodes with exactly defined types

VNet specificity: average specificity of its nodes
Price of Specificity (PoS) - Definition

▶ Cost$_\sigma$: cost under a given specificity $\sigma$(VNet)
▶ Cost$_0$: cost without specification constraints

Price of Specificity definition:

$$PoS = \frac{\text{Cost}_\sigma}{\text{Cost}_0}$$
PoS - Example

- Spec.: Spain + Italy ($\sigma = 1$)

- No specification ($\sigma = 0$)
PoS - Example

Cost metric: Number of hops

- Spec.: Spain + Italy ($\sigma = 1$)
  - 3 hops

- No specification ($\sigma = 0$)
  - 1 hop
PoS - Example

Cost metric: Number of hops

Spec.: Spain + Italy ($\sigma = 1$)
- 3 hops

No specification ($\sigma = 0$)
- 1 hop

$$PoS = \frac{3}{1} = 3$$
Overview

Introduction

- Embedding problem
- Specification, *PoS*
Overview

Introduction

- Embedding problem
- Specification, PoS

Upcoming

- Embedding algorithm
- Impact of different factors on the PoS
Optimal Algorithm

Constants:

Requests : $R$
Virtual Vertices : $V_v(r), r \in R$
Substrate Vertices : $V_s$
Virtual Edges : $E_v(r) \rightarrow V_v(r), r \in R$
Substrate Edges : $E_s : V_s \times V_s$
Unique : $uni\_checks : \forall (s_1, s_2) \in E_s : (s_2, s_1) \notin E_s$
Unique : $uni\_check_r : \forall r \in R, (v_1, v_2) \in E_v(r) : (v_2, v_1) \notin E_v(r)$
Substrate Edges: Virtual Edges: Embedding Cost: $\min \sum_{r \in R, (v_1, v_2) \in E_v(r), (s_1, s_2) \in E_s} f_{alloc}(r, v_1, v_2, s_1, s_2) + f_{alloc}(r, v_1, v_2, s_2, s_1)$

Variables:

Node Mapping : $n\_map(r, v, s) \in \{0, 1\}, r \in R, v \in V_v(r), s \in V_s$
Flow Allocation : $f_{alloc}(r, e, eb) \geq 0, r \in R, e \in E_v(r), eb \in EB_s$

Constraints:

Each Node Mapped : $\forall r \in R, v \in V_v(r) : \sum_{s \in V_s} n\_map(r, v, s) \cdot place(r, v, s) = 1$
Feasible : $\forall s \in V_s : \sum_{r \in R, v \in V_v(r)} n\_map(r, v, s) \cdot \vnd(r, v) \leq \snc(s)$
Guarantee Link Realization : $\forall r \in R, (v_1, v_2) \in E_v(r), s \in V_s : \sum_{(s_1, s_2) \in V_s \times V_s} f_{alloc}(r, v_1, v_2, s_1, s_2) =$
$\sum_{(s_1, s_2) \in V_s \times V_s} f_{alloc}(r, v_1, v_2, s_1, s_2) = vld(r, v_1, v_2) \cdot (n\_map(r, v_1, s) - n\_map(r, v_2, s))$
Realize Flows : $\forall (s_1, s_2) \in E_s : \sum_{r \in R, (v_1, v_2) \in E_v(r)} f_{alloc}(r, v_1, v_2, s_1, s_2) + f_{alloc}(r, v_1, v_2, s_2, s_1) \leq \scl(s_1, s_2)$

Objective function:

Minimize Embedding Cost : $\min : \sum_{r \in R, (v_1, v_2) \in E_v(r), (s_1, s_2) \in E_s} f_{alloc}(r, v_1, v_2, s_1, s_2) + f_{alloc}(r, v_1, v_2, s_2, s_1)$
Optimal Algorithm

Constants:
- Requests: \( R \)
- Substrate Vertices: \( V_s \)
- Virtual Vertices: \( V_v(r), r \in R \)
- Substrate Edges: \( E_s : V_s \times V_s \)

\[
\text{Embedding Cost: } \min \sum_{r \in R, (v_1,v_2) \in E_v(r), (s_1,s_2) \in E_s} f_{\text{alloc}}(r, v_1, v_2, s_1, s_2) + f_{\text{alloc}}(r, v_1, v_2, s_2, s_1)
\]

MIP (Mixed-integer program)
- Objective function: Minimize Link Cost
- Constraints to ensure feasibility
- Migration
- Different types of links
- Optimal embeddings

Objective function:
Minimize Embedding Cost: \( \min: \sum_{r \in R, (v_1,v_2) \in E_v(r), (s_1,s_2) \in E_s} f_{\text{alloc}}(r, v_1, v_2, s_1, s_2) + f_{\text{alloc}}(r, v_1, v_2, s_2, s_1) \)
Impact of Substrate Size

- 5-star VNet
- Node capacity of one
- Substrates created with *Igen topology generator*

![Graph showing the impact of substrate size on PoS and specificity](image)
Impact of Substrate Size

- 5-star VNet
- Node capacity of one
- Substrates created with Igen topology generator
Impact of Substrate Size

- 5-star VNet
- Node capacity of one
- Substrates created with Igen topology generator
Impact of Substrate Size

- 5-star VNet
- Node capacity of one
- Substrates created with \textit{Igen topology generator}
Impact of Node Capacity

- 5-star VNet
- Colocation allowed
- 100 nodes substrate created with Igen
Impact of Node Capacity

- 5-star VNet
- Colocation allowed
- 100 nodes substrate created with Igen
Impact of Node Capacity

- 5-star VNet
- Colocation allowed
- 100 nodes substrate created with Igen
Migration

- Substrate:

- Migration lowers average resource cost in general
- Depends on access policy
- Various impacts on PoS
Migration

- VNet:
- Substrate:
Impact of Migration

- Migration lowers average resource cost in general
- Depends on access policy
- Various impacts on PoS
Migration

- VNet:

- Substrate:
Impact of Migration

- Migration lowers average resource cost in general
- Depends on access policy
- Various impacts on PoS
## Summary

<table>
<thead>
<tr>
<th>Impact of</th>
<th>PoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substrate size</td>
<td></td>
</tr>
<tr>
<td>Node capacity</td>
<td></td>
</tr>
<tr>
<td>Migration</td>
<td></td>
</tr>
</tbody>
</table>

14 / 15
### Summary

<table>
<thead>
<tr>
<th>Impact of</th>
<th>PoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substrate size</td>
<td></td>
</tr>
<tr>
<td>Node capacity</td>
<td></td>
</tr>
<tr>
<td>Migration</td>
<td></td>
</tr>
<tr>
<td>Substrate load</td>
<td></td>
</tr>
</tbody>
</table>
### Summary

<table>
<thead>
<tr>
<th>Impact of</th>
<th>PoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substrate size</td>
<td>↓</td>
</tr>
<tr>
<td>Node capacity</td>
<td>↑</td>
</tr>
<tr>
<td>Migration</td>
<td>×</td>
</tr>
<tr>
<td>Substrate load</td>
<td>↑</td>
</tr>
<tr>
<td>VNet size</td>
<td>↑</td>
</tr>
</tbody>
</table>

Type distribution: 14 / 15
Summary

<table>
<thead>
<tr>
<th>Impact of</th>
<th>PoS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Substrate size</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Node capacity</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Migration</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Substrate load</strong></td>
<td></td>
</tr>
<tr>
<td><strong>VNet size</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Type distribution</strong></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- Impact of VNet specification on the embedding cost
- Optimal embeddings
- General embedding algorithm
- PoS, tool to adjust pricing and embedding (applied as a factor?)

Prototype (open source)
FLERD
Project website: www.net.t-labs.tu-berlin.de/~stefan/virtu.shtml
Conclusion

- Impact of VNet specification on the embedding cost
- Optimal embeddings
- General embedding algorithm
- PoS, tool to adjust pricing and embedding (applied as a factor?)

- Prototype (open source)
- Specification language FLERD
- Project website*

*www.net.t-labs.tu-berlin.de/~stefan/virtu.shtml