**Motivation**

Facilitating Management of Virtual Networks

- Economic aspects
  - Dynamic & efficient resource usage
  - New business fields and models
- Security aspects
  - Domain isolation
- Operational aspects
  - Abstraction
  - Out-of-band debugging
  - Potentially higher fault tolerance

$\Rightarrow$ Effective, economic management of Virtual Networks

**Scenario and Goal**

**Players**

- VNet User
- VNet Provider
- Infrastructure Provider

**Services**

- Infrastructure Provider: Provides Virtual Resources and Resource Control Interface
- VNet Provider: Assembles Virtual Networks
- VNet Operator: Operates, controls, manages virtual networks
- Service Provider: Service level customer support

**Online Migration Algorithm**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>MIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>opt</td>
<td>opt</td>
<td>opt</td>
<td>opt</td>
<td>opt</td>
<td>MIG</td>
</tr>
<tr>
<td>40 (A)</td>
<td>42 (A)</td>
<td>38 (A)</td>
<td>44 (A)</td>
<td>36 (A)</td>
<td></td>
</tr>
</tbody>
</table>

Strike balance between $\text{Cost}^\text{MIG}$ and $\text{Cost}^\text{MIG}_\text{acc}$

- Let $\beta = \max\{\text{Cost}_{\text{w},t}, \forall t \in V\}$
- Count $L_\alpha = \sum \text{Cost}_{\text{w},t}, \forall t \in V$
- When $L_\alpha \geq \beta$ for server location, end phase, and migrate to $\epsilon'$ with $L_\alpha < \beta$
- When $L_\alpha \geq \beta$ for $\forall \epsilon \in V$, end epoch $\tau$, and reset $L_\alpha = \beta$

1. Define $\text{Cost}(\epsilon)$, Define $\beta$ $\Rightarrow \forall \epsilon: \text{OPT}(\epsilon) \geq \beta$
2. $H_\alpha$ migrations expected $\Rightarrow H_\alpha + 1$ phases expected
3. $(\text{OPT} + 1) \Rightarrow \text{MIG}(\epsilon) \leq \beta H_\alpha + (\beta H_\alpha + 1) = \beta O(\log n)$
4. $(\text{OPT} + 1) \Rightarrow \text{Ratio} \rho \leq \frac{\beta O(\log n)}{\text{OPT}(\epsilon)} = O(\log n)$

**Server Migration Competitiveness**

Dynamic programming

1. $\text{OPT}(\epsilon) = \min_{(t-1)} [\text{OPT}(t-1) + \text{Cost}(\epsilon)]$
2. Remember predecessor $\epsilon_{t-1} \in V$
3. Optimal substructure property

**Optimal Offline Algorithm**

$\text{OPT}(\epsilon) = \min_{(t-1)} [\text{OPT}(t-1) + \text{Cost}(\epsilon)]$

$\Rightarrow \text{OPT}(\epsilon) = \sum_{\epsilon \in V} \text{Cost}(\epsilon)

VNO view:
- No knowledge of Substrate required
- SP requests latency reduction
- VNO changes virtual resource requirements
- VNO negotiates with VNP

**Pip view**
- No knowledge of VNet internal semantics required
- Receives updated requirements
- Initiates migration to effect latency drop

**Techniques**

- Competitive analysis
- Dynamic programming

**VNO view**

Initiates migration to effect latency drop

Receives updated requirements

- No knowledge of Substrate required

**Testbed**

- Distributed Virtual Network Testbed
- Proof-of-concept implementations