Privacy-Oriented Virtual Network Embedding

Leonardo Richter Bays
Rodrigo Ruas Oliveira
Luciana Salete Buriol
Marinho Pilla Barcellos
Luciano Paschoal Gaspary

May 2015

.inf INSTITUTO DE INFORMÁTICA UFRGS

Overview

Network Virtualization

- Enables the creation of virtual topologies on top of physical substrates
- Has been embraced by academic researchers and the Industry
- Key concerns: **efficient resource mapping** and **privacy**

Privacy

- Recent discovery of pervasive electronic surveillance has highlighted privacy concerns
- These concerns are even more exacerbated in network virtualization
- Security mechanisms must be considered in order to maintain a desired level of privacy

.inf INSTITUTO DE INFORMÁTICA UFRGS

Related Work

- Different methods have been used to solve the embedding problem:
  - Optimization Models – [Chowdhury et al., 2011; Alkmim et al., 2013]
  - Relaxations – [Chowdhury et al., 2011; Alkmim et al., 2013]
  - Heuristic approaches – [Yu et al., 2008; Cheng et al., 2011]
- We are not aware of previous attempts to reconcile efficient resource mapping and the satisfaction of security requirements

Objectives

- **Reconciling**:
  - **Efficient usage** of physical resources
  - Fulfillment of different levels of **confidentiality**:
    - End-to-end cryptography
    - Point-to-point cryptography
    - Non-overlapping networks
- Considering precise costs of security mechanisms

Proposed Optimization Model

- Based on **Integer Linear Programming**
- Inputs:
  - Physical and virtual topologies
  - Routers: throughput, cryptographic protocol support
  - Links: bandwidth
  - Locations
  - Costs related to cryptographic operations
  - Conflicting VNs
  - Previously embedded VNs

Proposed Optimization Model

Objective Function: Minimize overall bandwidth consumption:

$$min \sum_{(i,j) \in L^P} \sum_{r \in N^V, (k,l) \in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l}$$

Constraints:

- Fulfill capacity requirements
- Ensure proper router and link mapping
- Ensure desired level of confidentiality
- Fulfill location requirements
- Maintain previous mappings

Proposed Heuristic Algorithm

- Based on **Simulated Annealing**
- Same inputs and constraints as the optimization model
- Aims at minimizing overall bandwidth consumption

Proposed Heuristic Algorithm

- Generates an initial mapping and iteratively moves virtual routers attempting to improve the solution
- Allows the generation of unfeasible solutions
  - Such solutions are penalized in order to encourage feasible ones
- Runs until the best found solution is close enough to optimality or a maximum number of iterations has been reached

Proposed Heuristic Algorithm

Algorithm Proposed Heuristic Algorithm

Proposed Heuristic Algorithm

Algorithm Proposed Heuristic Algorithm

```
s ← generateInitialSolution
c ← evaluateSolution(s)
s^best ← s; c^best ← c
```

Proposed Heuristic Algorithm

Algorithm Proposed Heuristic Algorithm

$s \leftarrow generateInitialSolution$
$c \leftarrow evaluateSolution(s)$
$s^{best} \leftarrow s$; $c^{best} \leftarrow c$
$k \leftarrow 0$
while $k < k^{max}$ **and** $c > c^{max}$ **do**
end while

Proposed Heuristic Algorithm

Algorithm Proposed Heuristic Algorithm

$s \leftarrow generateInitialSolution$
$c \leftarrow evaluateSolution(s)$
$s^{best} \leftarrow s$; $c^{best} \leftarrow c$
$k \leftarrow 0$
while $k < k^{max}$ **and** $c > c^{max}$ **do**
    $s' \leftarrow generateNeighbor(s)$
    $c' \leftarrow evaluateSolution(s')$
end while

Proposed Heuristic Algorithm

Algorithm Proposed Heuristic Algorithm

$s \leftarrow generateInitialSolution$

$c \leftarrow evaluateSolution(s)$

$s^{best} \leftarrow s;\ c^{best} \leftarrow c$

$k \leftarrow 0$

while $k < k^{max}$ **and** $c > c^{max}$ **do**

    $s' \leftarrow generateNeighbor(s)$

    $c' \leftarrow evaluateSolution(s')$

    $t \leftarrow temperature(k, k^{max})$

    if $probability(c, c', t) > random[0, 1)$ **then**

        $s \leftarrow s';\ c \leftarrow c'$

    end if

end while

Proposed Heuristic Algorithm

Algorithm Proposed Heuristic Algorithm

$s \leftarrow \textit{generateInitialSolution}$
$c \leftarrow \textit{evaluateSolution}(s)$
$s^{best} \leftarrow s;\ c^{best} \leftarrow c$
$k \leftarrow 0$
while $k < k^{max}$ **and** $c > c^{max}$ **do**
  $s' \leftarrow \textit{generateNeighbor}(s)$
  $c' \leftarrow \textit{evaluateSolution}(s')$
  $t \leftarrow \textit{temperature}(k, k^{max})$
  if $\textit{probability}(c, c', t) > \textit{random}[0, 1)$ **then**
    $s \leftarrow s';\ c \leftarrow c'$
  end if
  if $c < c^{best}$ **and** $\textit{isFeasible}(s)$ **then**
    $s^{best} \leftarrow s;\ c^{best} \leftarrow c$
  end if
  $k \leftarrow k + 1$
end while

Evaluation – Fixed Parameters

- General parameters:
  - 250 time slots
  - On average, 5 requests arrive per slot (Poisson)
  - Average duration of 5 slots (exponential)
- Heuristic algorithm parameters:
  - Maximum number of iterations: 5,000
  - Maximum overhead factor: 3
  - Penalty factor: 100

Evaluation – Fixed Parameters

- Physical network:
  - Routers: 10 Gbps throughput
  - Distributed among 16 locations
  - 95% support cryptographic protocols
  - Links: 1–10 Gbps
- Virtual networks:
  - Two routers with location constraints
  - 35% of VNs do not require cryptography
  - 35% of VNs require end-to-end cryptography
  - 30% of VNs require point-to-point cryptography
  - 5% of VNs conflict with a previously embedded VN

.inf INSTITUTO DE INFORMÁTICA UFRGS

Evaluation – Variable Parameters

- Network sizes:
  - Physical network of size 100, VN sizes ranging from 2 to 5
  - Physical network of size 500, VN sizes ranging from 2 to 10
- Cryptographic Algorithms:
  - AES-128
  - AES-256

Results

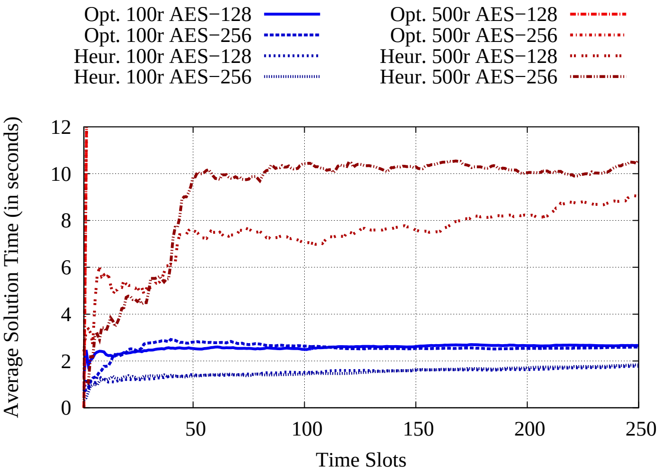


Figure: Time needed to find the accepted solution.

Results

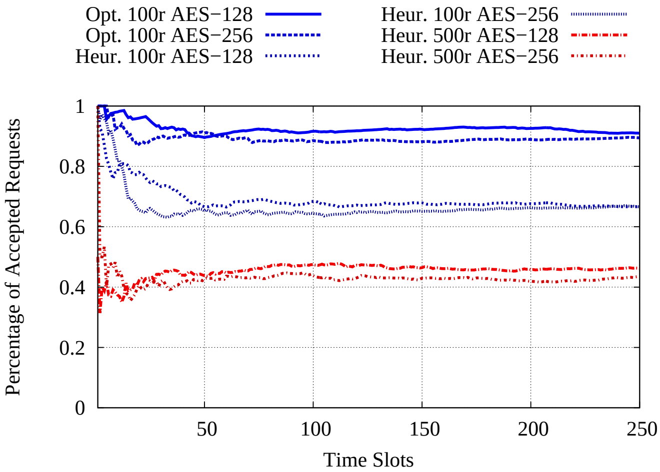
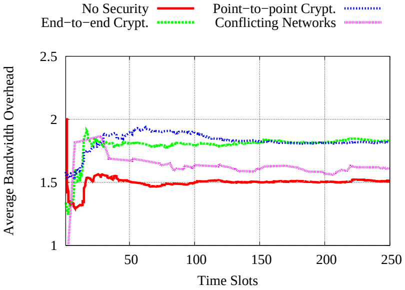
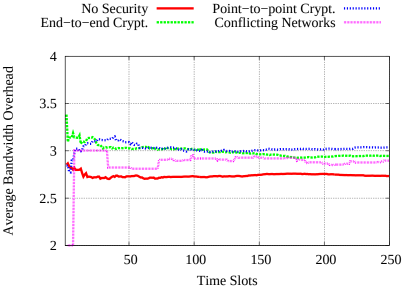


Figure: Acceptance rate in all completed experiments.

Results



(a) Opt. 100r AES-256



(b) Heur. 100r AES-256

Figure: Average bandwidth overhead needed to embed requests of different kinds.

Conclusions

- Optimization model produces adequate results in a timely manner for networks of limited size
- Heuristic algorithm scales to larger networks without a significant increase in solution time
- If desired, the heuristic algorithm may lead to more precise results through parameterization

Ongoing Work

- Further improvements to the heuristic algorithm
  - First-improvement-based local search
  - Multiple moves per temperature change
  - Logarithmic cooling schedule
- Analysis of the impact of topological factors on the VN embedding process
- SDN integration case study

Privacy-Oriented Virtual Network Embedding

Leonardo Richter Bays
Rodrigo Ruas Oliveira
Luciana Salete Buriol
Marinho Pilla Barcellos
Luciano Paschoal Gaspary

`lrbays@inf.ufrgs.br`
`http://inf.ufrgs.br/~lrbays`

INSTITUTO DE INFORMÁTICA UFRGS

References

Gustavo P. Alkmim, Daniel M. Batista, and Nelson L. S. Fonseca. Mapping virtual networks onto substrate networks. *Journal of Internet Services and Applications*, 3(4), 2013. ISSN 1869-0238. doi: 10.1186/1869-0238-4-3.

Xiang Cheng, Sen Su, Zhongbao Zhang, Hanchi Wang, Fangchun Yang, Yan Luo, and Jie Wang. Virtual network embedding through topology-aware node ranking. *SIGCOMM Computer Communication Review*, 41(2), 2011. ISSN 0146-4833. doi: 10.1145/1971162.1971168.

M. Chowdhury, M. R. Rahman, and R. Boutaba. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Transactions on Networking*, PP(99), 2011. ISSN 1063-6692. doi: 10.1109/TNET.2011.2159308.

Minlan Yu, Yung Yi, Jennifer Rexford, and Mung Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *SIGCOMM Computer Communication Review*, 38(2), 2008. ISSN 0146-4833. doi: 10.1145/1355734.1355737.

Constraints

Constraints C1 and C3: Fulfill capacity requirements:

$$\sum_{r \in N^V, j \in R^V} T^V_{r,j} W^R_{r,j} A^R_{i,r,j} \leq T^P_i \qquad \forall i \in R^P \quad \text{(C1)}$$

$$\sum_{r \in N^V, (k,l) \in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l} \leq B^P_{i,j} \qquad \forall (i,j) \in L^P \quad \text{(C3)}$$

Constraints

Constraints C2, C5, and C6: Ensure proper router and link mapping:

$$\sum_{j \in R^V} A^R_{i,r,j} \leq 1 \qquad \forall i \in R^P, r \in N^V \quad \text{(C2)}$$

$$\sum_{i \in R^P} A^R_{i,r,j} = 1 \qquad \forall r \in N^V, j \in R^V \quad \text{(C5)}$$

$$\sum_{j \in R^P} A^L_{i,j,r,k,l} - \sum_{j \in R^P} A^L_{j,i,r,k,l} = A^R_{i,r,k} - A^R_{i,r,l} \qquad \forall r \in N^V, (k,l) \in L^V, i \in R^P \quad \text{(C6)}$$

Constraints

Constraints C4, C7, and C8: Ensure desired level of security:

$$K^V_{r,j} A^R_{i,r,j} \leq K^P_i \qquad \forall i \in R^P, r \in N^V, j \in R^V \quad \text{(C4)}$$

$$\sum_{q \in N^V, k \in R^V} A^R_{i,q,k} + \sum_{r \in N^V, l \in R^V} A^R_{i,r,l} \leq 1 \qquad \forall q, r \in X, i \in R^P \quad \text{(C7)}$$

$$\left\lceil \frac{\sum_{q \in N^V, (k,l) \in L^V} A^L_{i,j,q,k,l}}{|L^P|} \right\rceil + \left\lceil \frac{\sum_{r \in N^V, (o,p) \in L^V} A^L_{i,j,r,o,p}}{|L^P|} \right\rceil \leq 1 \qquad \forall q, r \in X, (i,j) \in L^P \quad \text{(C8)}$$

Constraints

Constraints C9, C10, and C11: Fulfill location requirements and maintain previous mappings:

$$jA^{R}_{i,r,k} = lA^{R}_{i,r,k} \qquad \forall (i,j) \in S^{P}, r \in N^{V}, (k,l) \in S^{V} \quad \text{(C9)}$$

$$A^{R}_{i,r,j} = E^{R}_{i,r,j} \qquad \forall (i,r,j) \in E^{R} \quad \text{(C10)}$$

$$A^{L}_{i,j,r,k,l} = E^{L}_{i,j,r,k,l} \qquad \forall (i,j,r,k,l) \in E^{L} \quad \text{(C11)}$$

Constraints

Constraint C8: Linearization:

$$Y_{q,r,i,j} \geq \frac{\sum_{q \in N^V,(k,l) \in L^V} A^L_{i,j,q,k,l}}{|L^P|} \qquad \forall q, r \in X, (i,j) \in L^P \quad \text{(C8.1)}$$

$$Z_{q,r,i,j} \geq \frac{\sum_{r \in N^V,(o,p) \in L^V} A^L_{i,j,r,o,p}}{|L^P|} \qquad \forall q, r \in X, (i,j) \in L^P \quad \text{(C8.2)}$$

$$Y_{q,r,i,j} + Z_{q,r,i,j} \leq 1 \qquad \forall q, r \in X, (i,j) \in L^P \quad \text{(C8.3)}$$

INSTITUTO DE INFORMÁTICA UFRGS

Heuristic Algorithm

- Evaluation Function:
  - If all constraints are satisfied:

$$\sum_{(i,j)\in L^P} \sum_{r\in N^V,(k,l)\in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l}$$

  - If any constraints are not satisfied:

$$\gamma \, \kappa \sum_{(i,j)\in L^P} \sum_{r\in N^V,(k,l)\in L^V} B^V_{r,k,l} W^L_r A^L_{i,j,r,k,l}$$

γ – severity of the applied penalty
κ – number of unsatisfied constraints

Heuristic Algorithm

- Stop criteria:
  - Maximum number of iterations has been reached; or
  - Bandwidth consumption of the best found solution is sufficiently close to optimality:

$$e \leq \beta \sum_{r \in N^V, (k,l) \in L^V} B^V_{r,k,l} W^L_r$$

β – maximum bandwidth overhead

Heuristic Algorithm

- Probability of moving to a neighbor solution:
  - If it is **better**: 1.0
  - If it is **worse**:

$$exp(\frac{e/enew - 1}{temperature})$$

$temperature = 1 - \frac{k}{kmax}$

Dijkstra's Algorithm

Weight of a link (i, j):

$$1 + \sum_{r \in N^V, (k,l) \in L^V} E^L_{i,j,r,k,l}$$