

# On Orchestrating Virtual Network Functions

---

Presented @ CNSM 2015

Md. Faizul Bari, Shihabur Rahman Chowdhury, and Reaz Ahmed, and Raouf Boutaba

David R. Cheriton School of Computer science

University of Waterloo



# Outline

- Background
- Motivation
- Problem Statement
- Proposed Solutions
- Evaluation
- Conclusion



# Background

- Traditional hardware middleboxes are:
  - Expensive
  - Proprietary
  - Vertically integrated
- Network Function Virtualization (NFV):
  - Packet processed by software middleboxes or Virtualized Network Functions (VNFs)
- Service Function Chaining
  - Traffic flows through an ordered sequence of middleboxes
    - Example:
      - Firewall → IDS → Proxy
      - Traffic Analyzer → Firewall → Video Optimizer



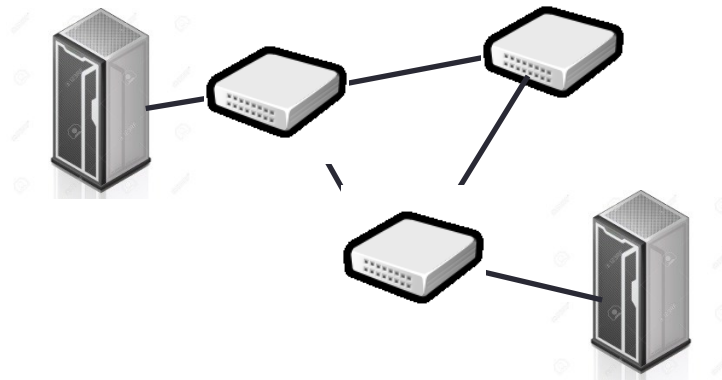
# Motivation

- Hardware middleboxes are:
  - Provisioned for ***peak traffic***
  - Attached at ***fixed locations*** in the network
  - Traffic is steered through these attachment points
    - Error-prone configuration process
- In case of VNFs:
  - VNF can be allocated ***based on demand***
  - Network attachment points ***are not fixed***
  - VNFs can be deployed on any compute server
  - Both VNF location and traffic steering can be optimized
    - Autonomic configuration
    - Reduction in OPEX



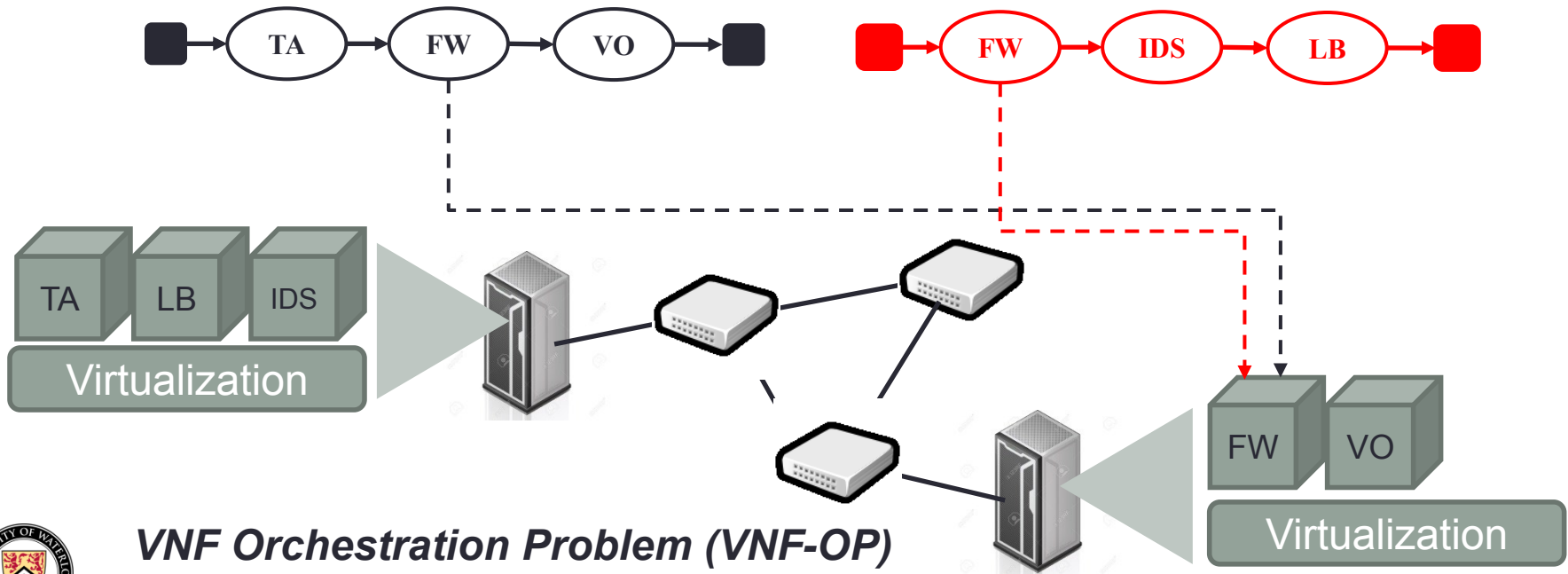
# Problem Statement

- Given
  - A set of VNF chain requests
  - Physical resource status



# Problem Statement (cont.)

- We need to decide
  - **How many** VNF instances (VM, container) to deploy?
  - **Where** to place them?
  - **Which VNF** (from chain) should be assigned where?
  - **How to route** traffic between the VNFs?



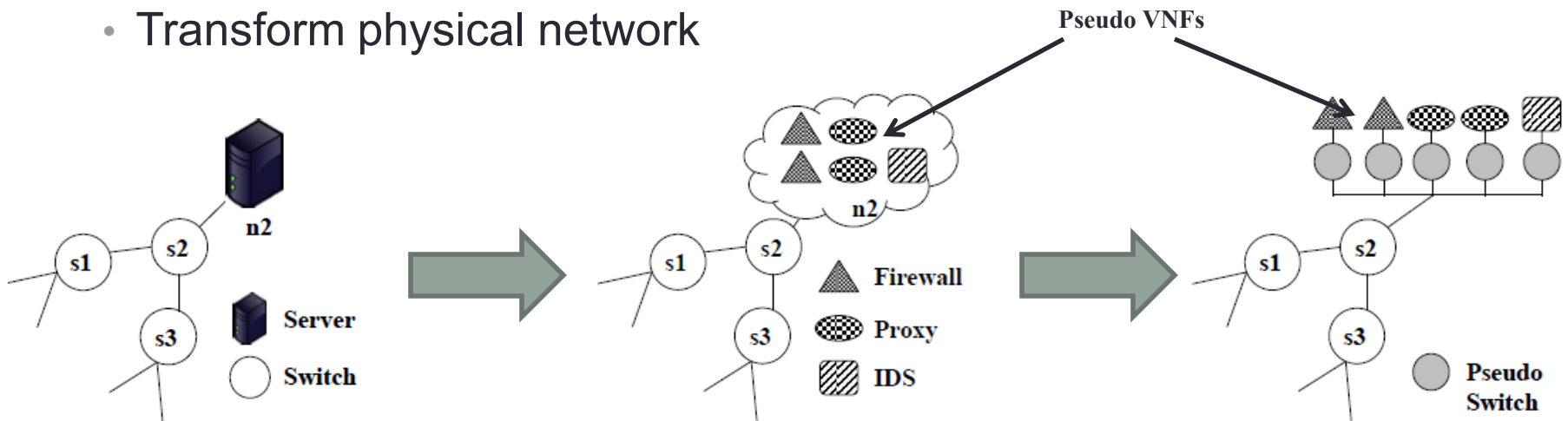
# Proposed Solution

- VNF-OP is a combination of three problems:
  - Allocating resource for VMs/containers
  - Assigning chain VNFs to these VMs
  - Finding routing paths for the chains
- Mathematical formulation is difficult:
  - Joint optimization results in quadratic constraints
  - Takes a long time to solve even for small problem instances



# Proposed Solution: Optimal (cont.)

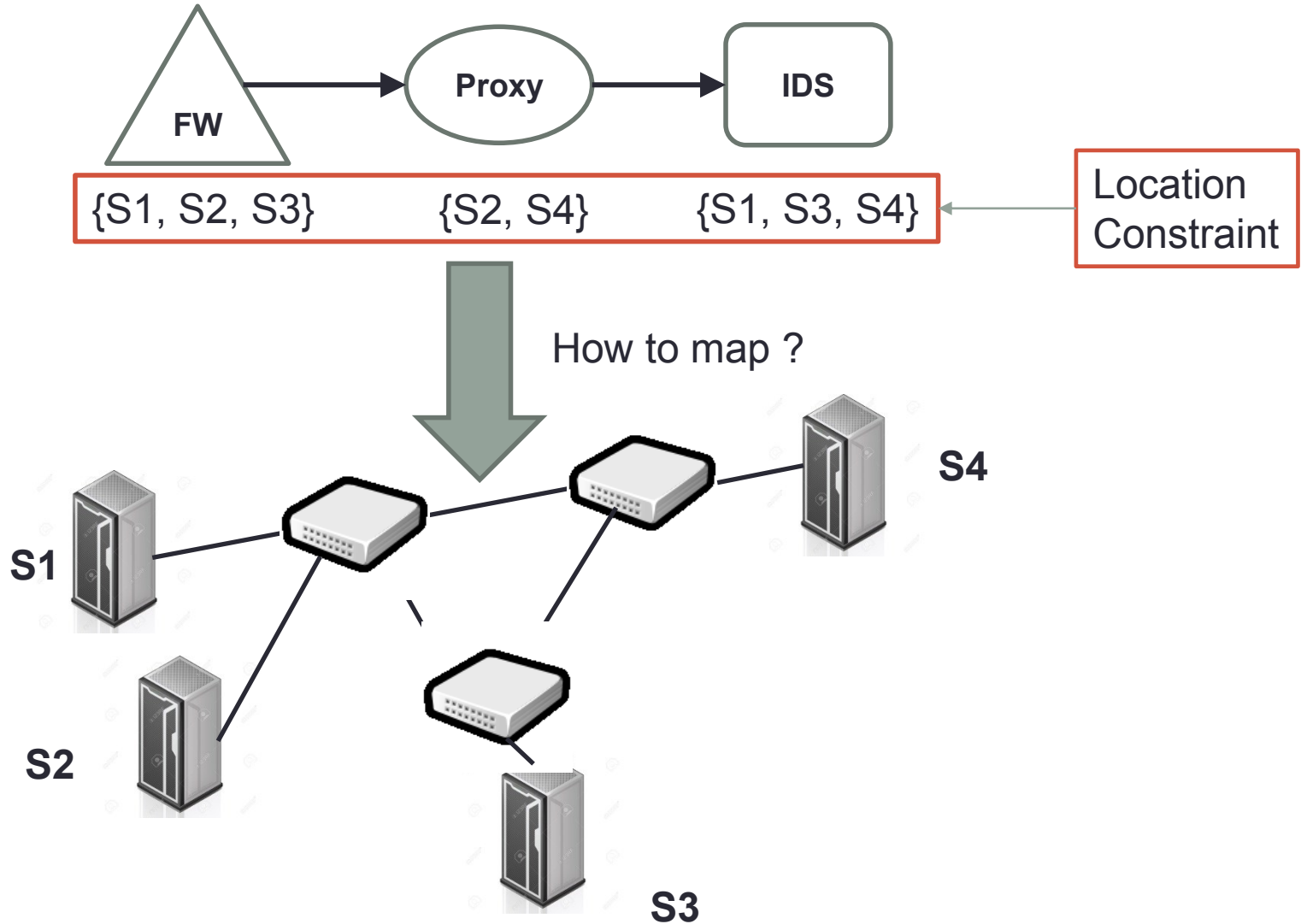
- Our approach:
  - Transform physical network



- Can be formulated as an ILP
- Much faster than implementation with quadratic constraints

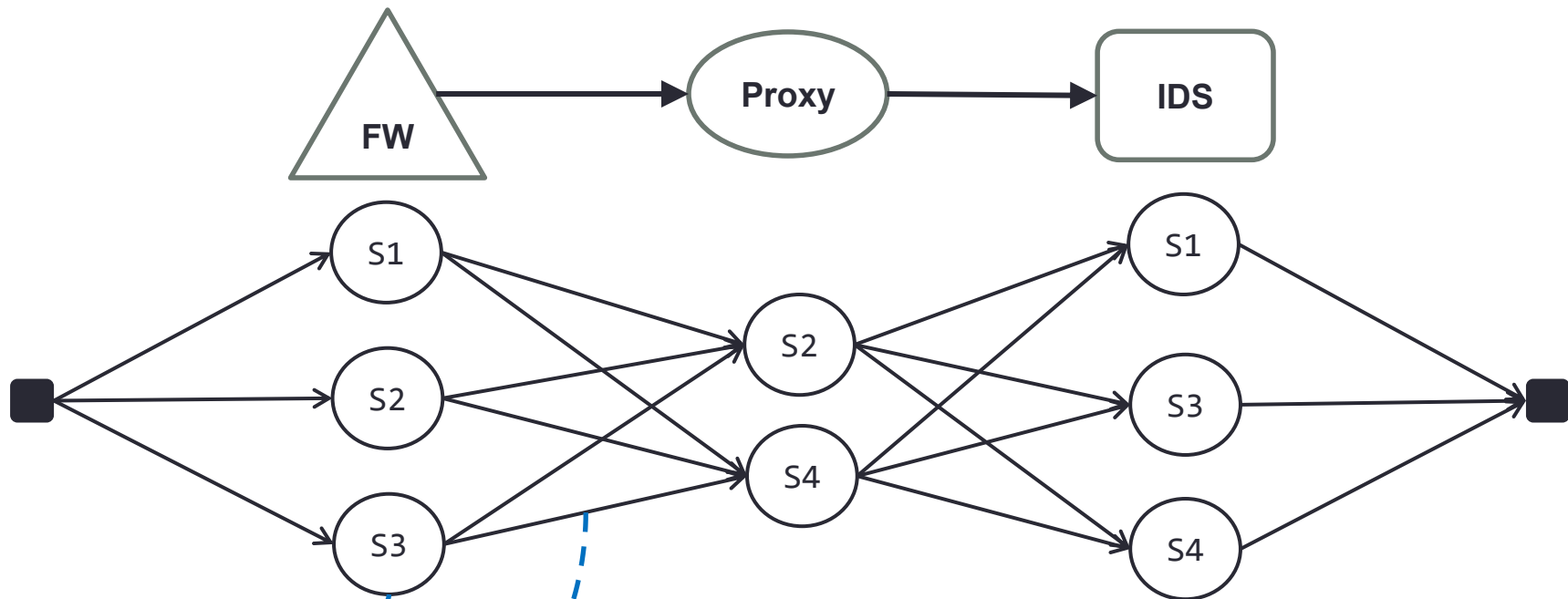


# Proposed Solution: Heuristic



# Proposed Solution: Heuristic (cont.)

Create a multi stage graph (one stage for each VNF) as follows:

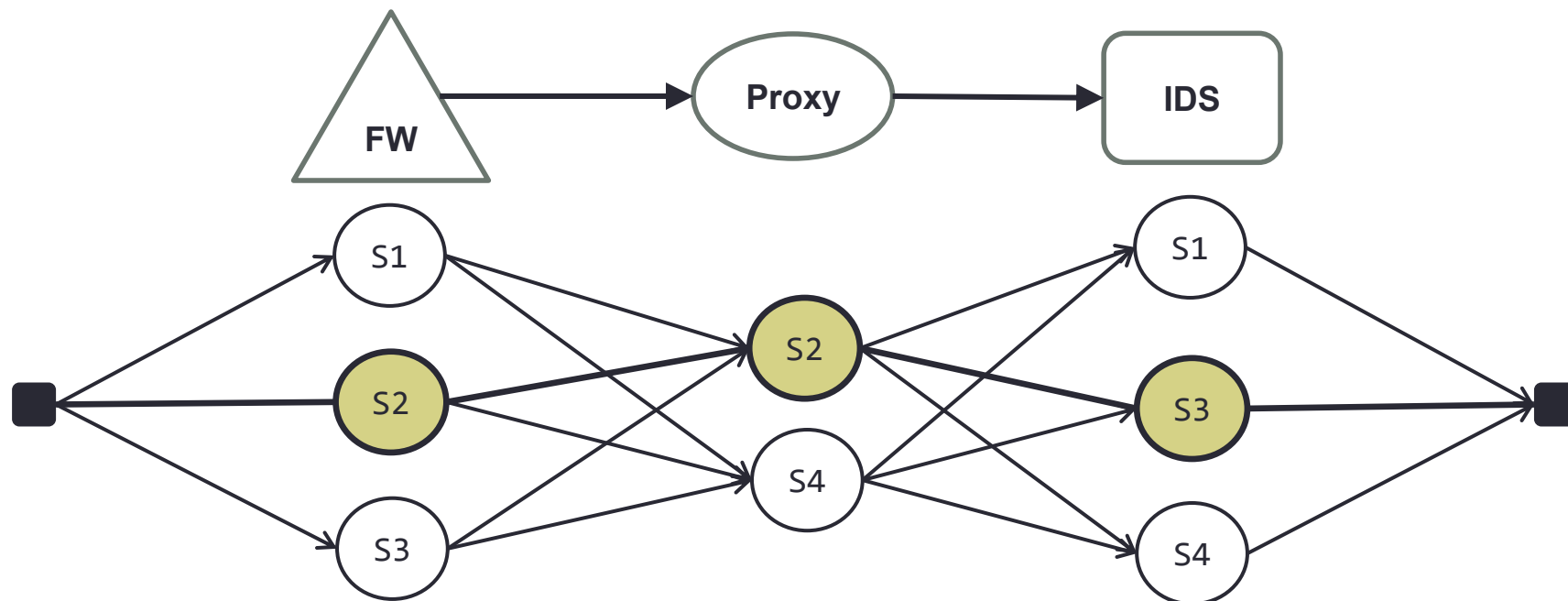


Possible locations  
for placing FW

Cost of placing Proxy at S2  
If FW is placed at S3

- **Objective: Find a path from left most to right most stage that has minimum cost**
  - **Select exactly one node at each stage**
- **Similar to assigning tags to an unknown sequence of observations based on known cost function**

# Proposed Solution: Heuristic (cont.)



- **Objective: Find a path from left most to right most stage that has minimum cost**
  - **Select exactly one node at each stage**
- **Similar to assigning tags to an unknown sequence of observations based on known cost function**

- **Solution: Use Viterbi algorithm to find the minimum cost path**
- **Viterbi is widely used in pattern recognition to assign tags to unknown sequence of observations.**

# Evaluation: Setup

- Two network topologies:
  - Internet2 research network (12 nodes, 15 links)
  - A university data center topology (23 nodes, 42 links)
- Server energy consumption data collected from Intel datasheet
- Hardware middlebox energy consumption data collected from a manufacturer
- Traffic traces
  - Traffic matrix from Internet2 network
  - Data center traffic trace from [1]

[1] T. Benson et al. Network traffic characteristics of data centers in the wild. ACM IMC '10



# Evaluation: Results

- **4x** OPEX reduction by VNFs compared to hardware middleboxes
- Our heuristic produces solutions that are within **1.3x** the optimal solution
- Heuristic is faster than the optimal
  - **65x** for Internet2
  - **3500x** for DC network



# Conclusion

- Proposed two solutions to the VNF orchestration problem:
  - CPLEX based optimal solution
  - Heuristic for larger networks
- Experiments suggest that NFV can reduce OPEX by a factor of 4 over hardware middleboxes
- Proposed heuristic produces near-optimal solutions in a fraction of the time required for the CPLEX program



# QUESTIONS?

---

# Viterbi Algorithm

