On the Safety and Efficiency of Virtual Firewall Elasticity Control

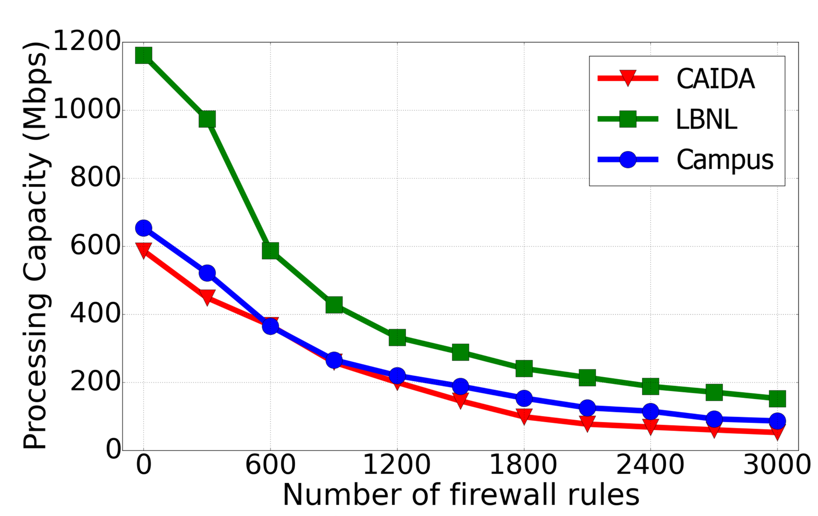
Hongda Li[1], Juan Deng[1], Hongxin Hu[1], Kuang-Ching Wang[1]
Gail-Joon Ahn[2], Ziming Zhao[2] and Wonkyu Han[2]
[1]Clemson University and [2]Arizona State University

Motivation

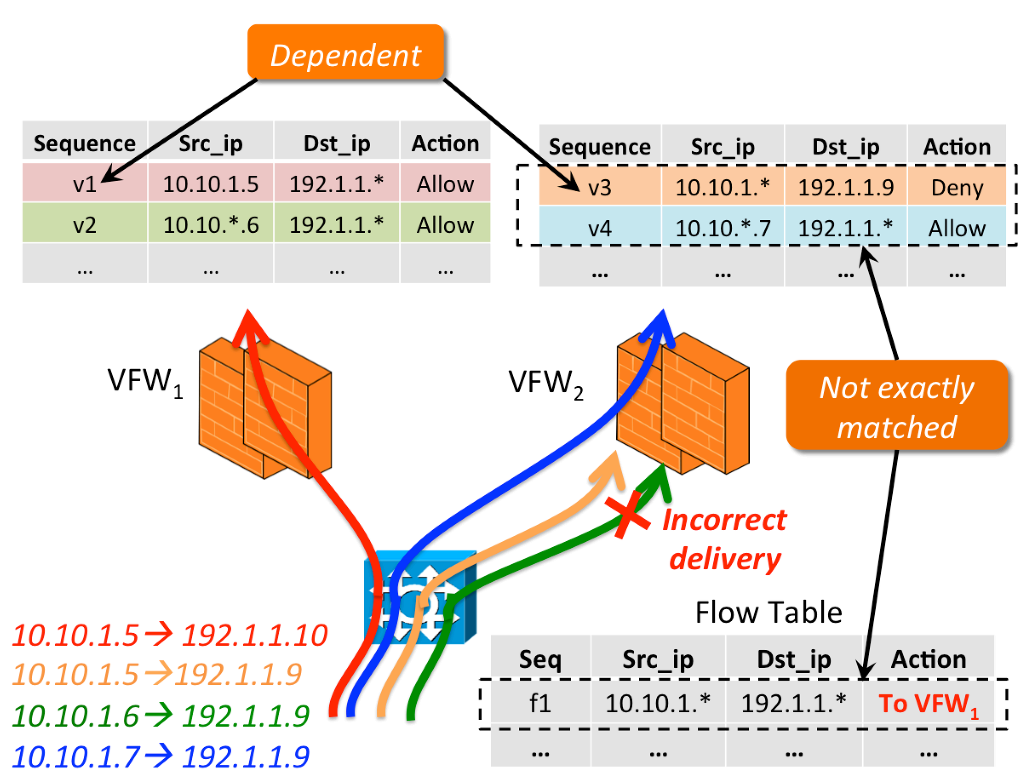
- ❖ Traditional Hardware-based Firewall
  - ➢ Fixed location & constant capacity
- ❖ New Requirements
  - ➢ Virtualized environments
    - ✓ Perimeter is blur & fluid
    - ✓ Services need migration often
  - ➢ Significant traffic volume variation
    - ✓ Elastic capacity
- ❖ New Trends
  - ➢ NFV: create and destroy software instances dynamically
  - ➢ SDN: dynamic traffic steering

NFV ✚ SDN ➜ Virtual Firewall

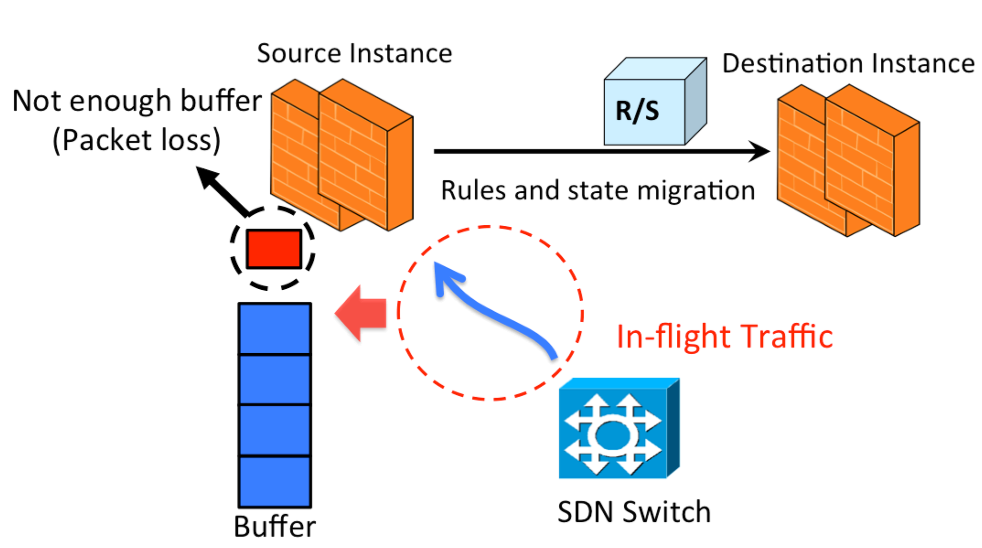
- ❖ Virtual Firewall Elastic Scaling
  - ➢ Overload → elastic scaling out
  - ➢ Underload → elastic scaling in
- ❖ Challenges to achieve *safe*, *efficient* and *optimal* virtual firewall scaling
  - ➢ Split or copy firewall policies?



- ➢ **Semantic consistency & correct flow update**



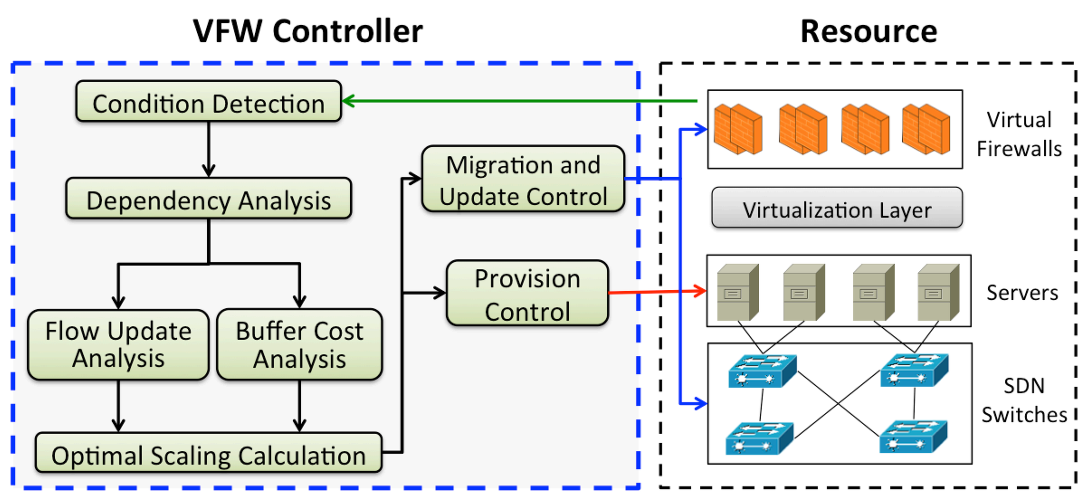
- ➢ **Buffer overflow avoidance**
  - ✓ Prior work assumes unlimited buffer size



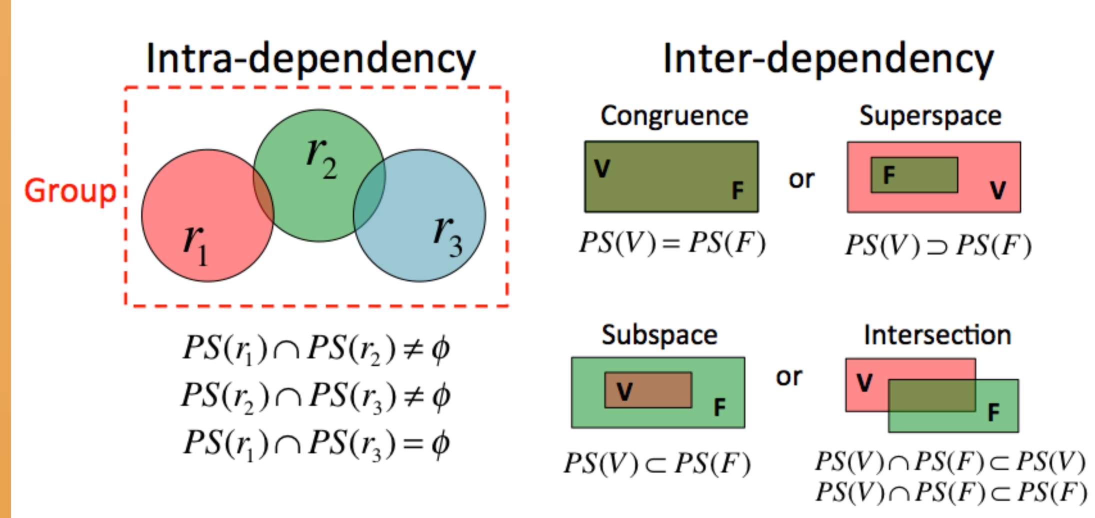
- ➢ **Optimal scaling**

Satisfy SLAs | *Minimize Update* | *Avoid Buffer Overflow*

Our Approach

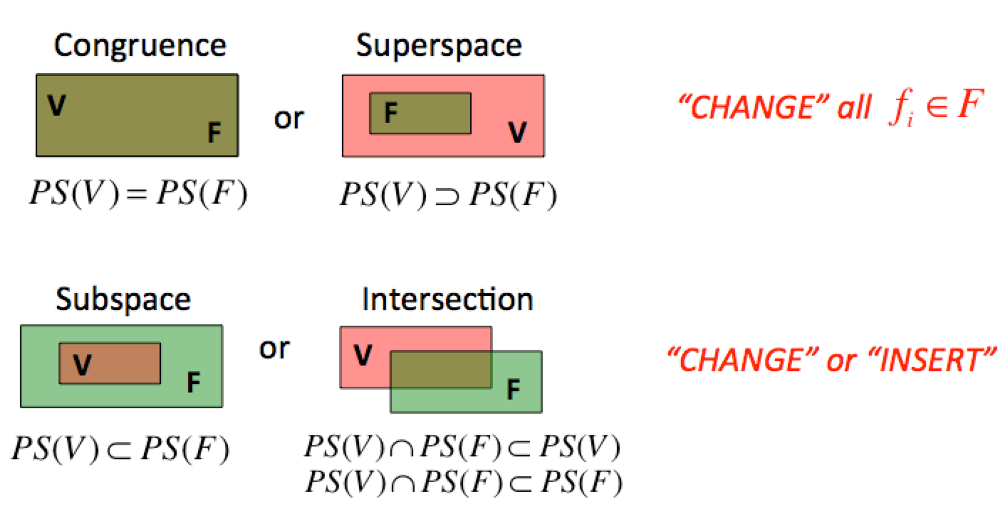


- ❖ Core Components of VFW Controller
  - ➢ Dependency Analysis
  - ➢ Flow Update Analysis
  - ➢ Buffer Cost Analysis
  - ➢ Optimal Scaling Calculation
- ❖ **Dependency Analysis**
  - ➢ Reasons to analyze dependencies
    - ✓ Intra-dependency for firewall rule migration
    - ✓ Inter-dependency for flow rule update



- ➢ Group-based firewall rule migration to ensure semantic consistency
- ❖ **Flow Update Analysis**
  - ➢ Update operation
    - ✓ CHANGE existing flow rules' actions
    - ✓ INSERT a new flow rule in front of an existing flow rule

***V**: firewall rule group to be migrated*
F**: flow rule group inter-dependent with **V



- ➢ Update cost
  - ✓ Number of new flow rules inserted
- ❖ **Buffer Cost Analysis**
  - ➢ d_1,d_2 and d_3 are transmission delays
  - ➢ b_1 and b_2 are average processing time per packet
  - ➢ λ_i is the traffic rate of f_i

$$\beta = (\sum \lambda_i) \times \{d_1 + d_3 - d_2 + b_1 + b_2\}$$

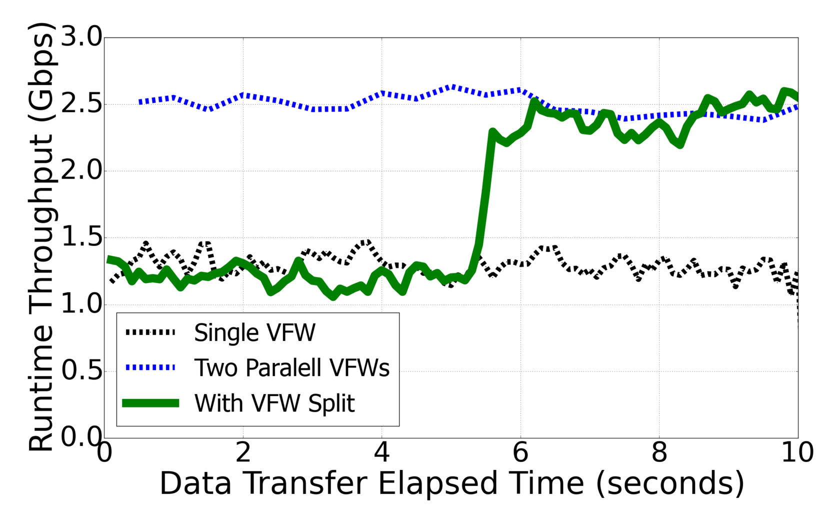
- ❖ **Optimal Scaling Calculation**
  - ➢ Scaling-out: least new instances
    - ✓ three-step heuristic
  - ➢ Scaling-in: most merged instances
    - ✓ integer linear programming

Implementation

- ❖ We implemented VFW Controller in real NFV/SDN platforms
  - ➢ Xen-4.4.1, ClickOS
  - ➢ Floodlight, Open vSwitch
  - ➢ Simple stateful firewall: new Click elements
  - ➢ VFW Controller: Hassel Library
  - ➢ Testbed: CloudLab (https://www.cloudlab.us/)
  - ➢ **Source code available**:
    - ✓ https://www.cloudlab.us/p/SeNFV/Firewall-VLANs

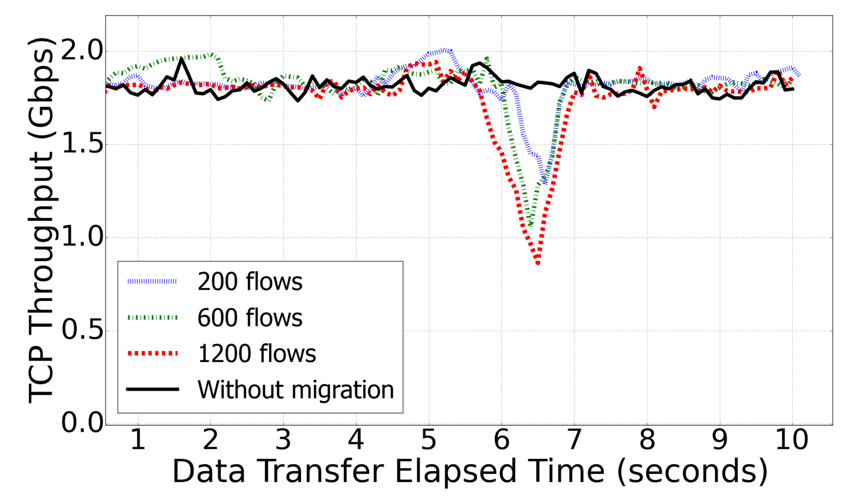
Evaluation

- ❖ Evaluation of group size based on real-world firewall policies
  - ➢ Largest firewall group contains 18 rules
- ❖ Capability to quickly scale
  - ➢ Scale in less than *1 sec* for 400 firewall rules



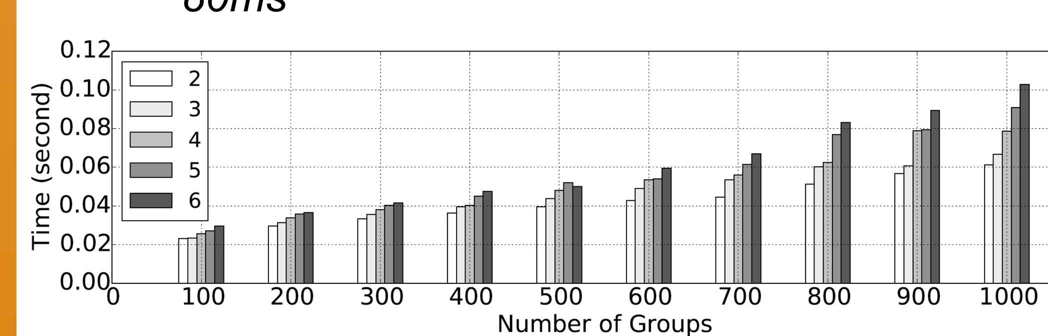
VFW split with TCP flow overload

- ❖ Migration impact on throughput
  - ➢ Quickly recover from migration
  - ➢ TCP connection preserved



Migration impact on TCP throughput

- ❖ Performance of optimal scaling calculation
  - ➢ 6 new instances, 1000 firewall rule groups in *110ms*
  - ➢ 100 underloaded virtual firewall instances in *80ms*



Optimal scaling calculation for scaling-out

Publication

❖ Deng J, Li H, Hu H, Wang KC, Ahn GJ, Zhao Z, Han W . "*On the Safety and Efficiency of Virtual Firewall Elasticity Control*" (NDSS 2017)