Distributed Network Service Policy Enforcement

Jun Li

Contributions from my students, Xiang Wang and Xiaohe Hu

Outline

- Background
- Related Work
- Policy Space Analysis
- Policy Enforcement with PSA

Orthogonal Perspective

- Forwarding vs. Service

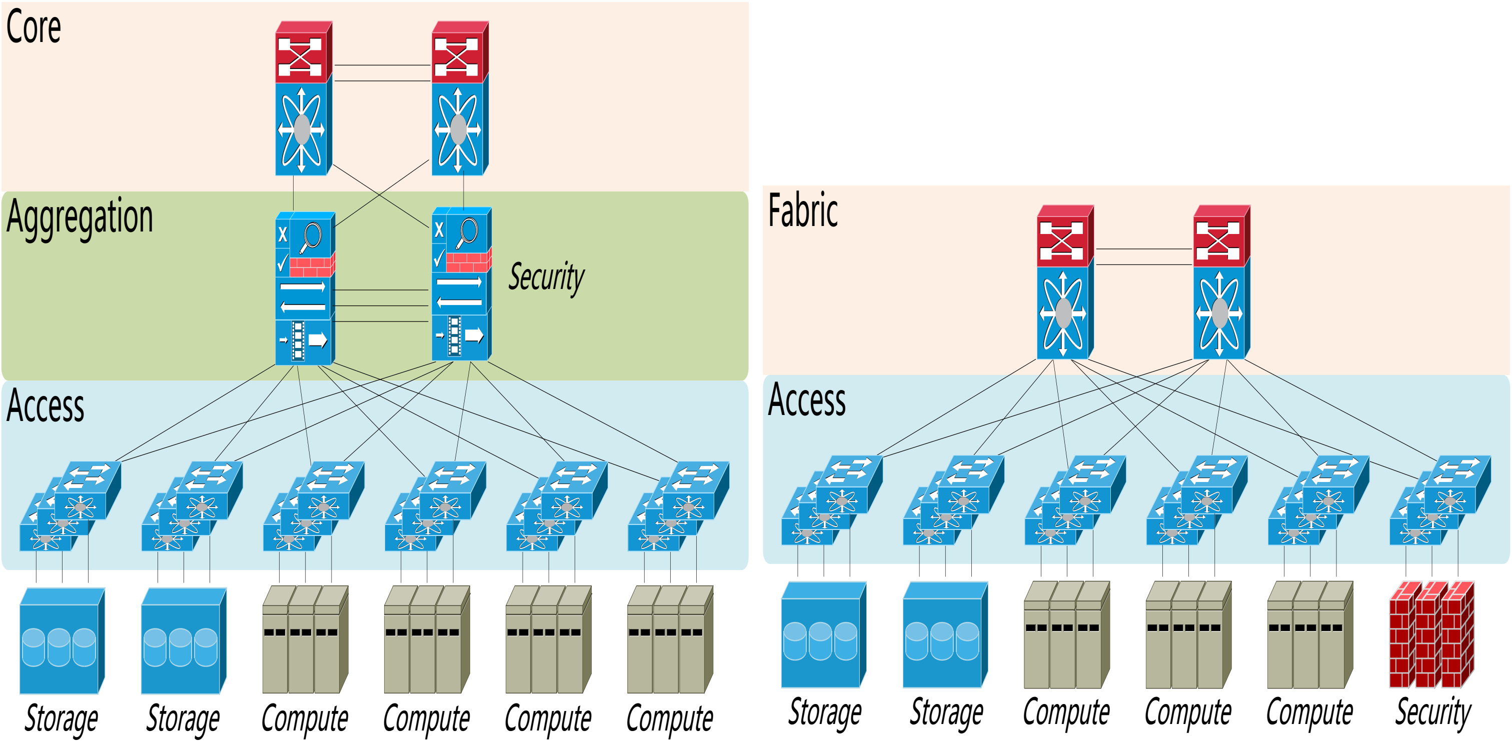
	Forwarding	Service
Task	Delivery	Security, Measurement, Optimization
Logical Object	Packet	Flow
Physical Object	L2 ~ L3, Header	L3 ~ L7, Header + Payload
Basis	Topology	Resource/Policy
State	Stateless	Stateful
Manner	Local autonomy	Global governance
Device	Switch/Router	Middlebox
Algorithm	Routing origination, Routing lookup	Packet classification, Pattern matching, AppID, Traffic management

Outline

- Background
- Related Work
- Policy Space Analysis
- Policy Enforcement with PSA

Data Center Network (DCN)

- Virtualization & Multi-tenancy

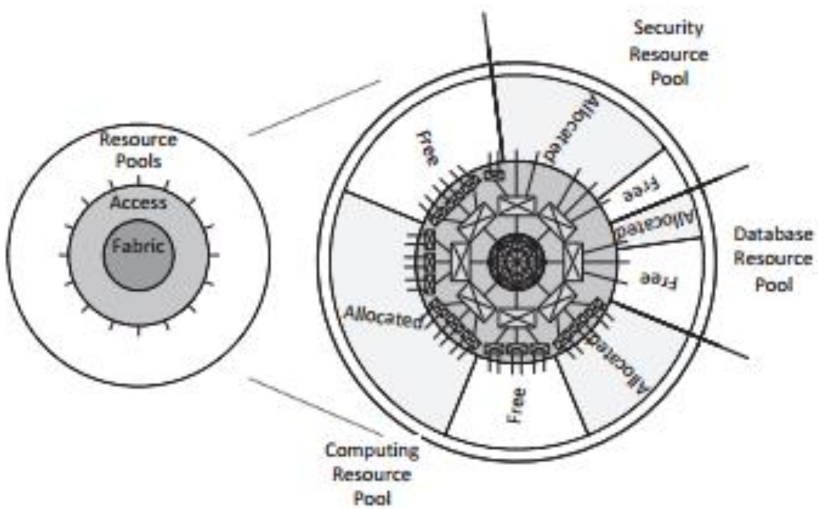


Traditional DCN architecture

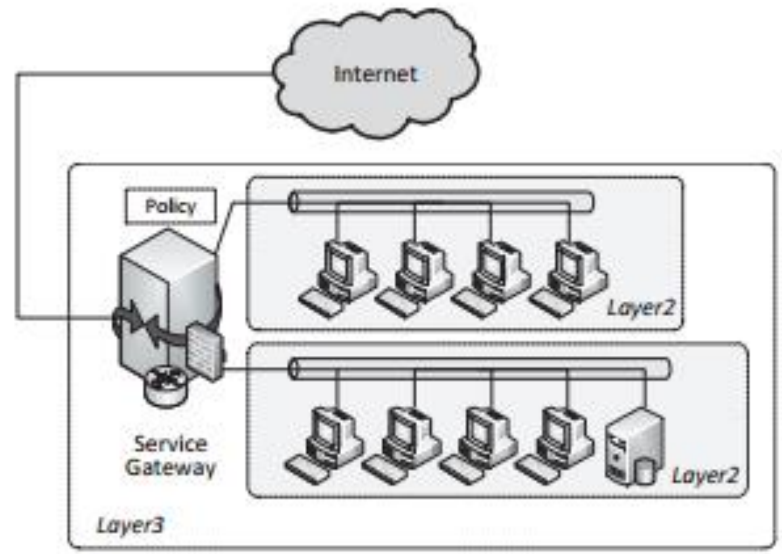
Cloud DCN architecture

Cloud DCN

- Logical View vs. Tenant View



Operator view for orchestration



Tenant view for provision

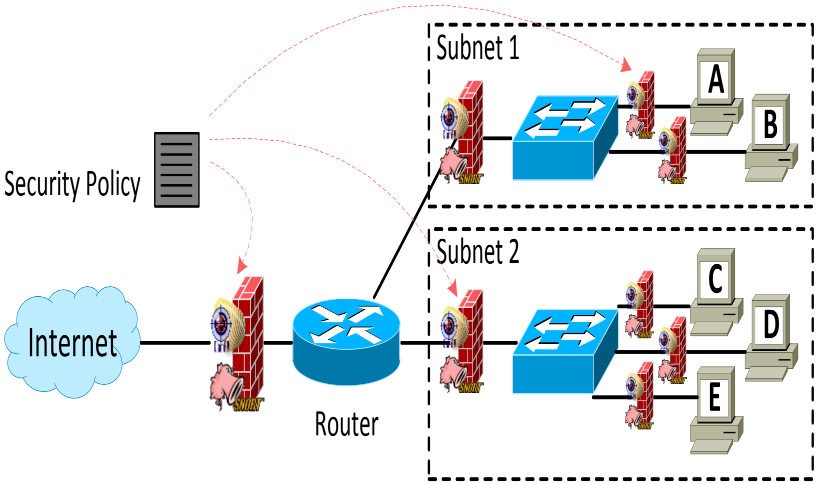
LiveCloud

[X. Wang et al., CouldCom 2012]

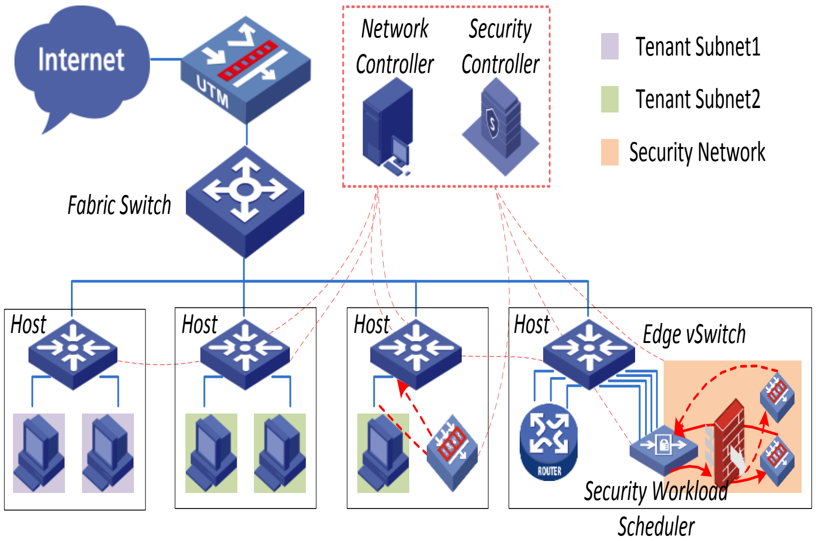
Service in Cloud DCN

- Network Service Mechanism
  - Share vs. Aggregation
  - Software vs. Hardware

Tualatin
[X. Wang et al., ICCCN 2014]



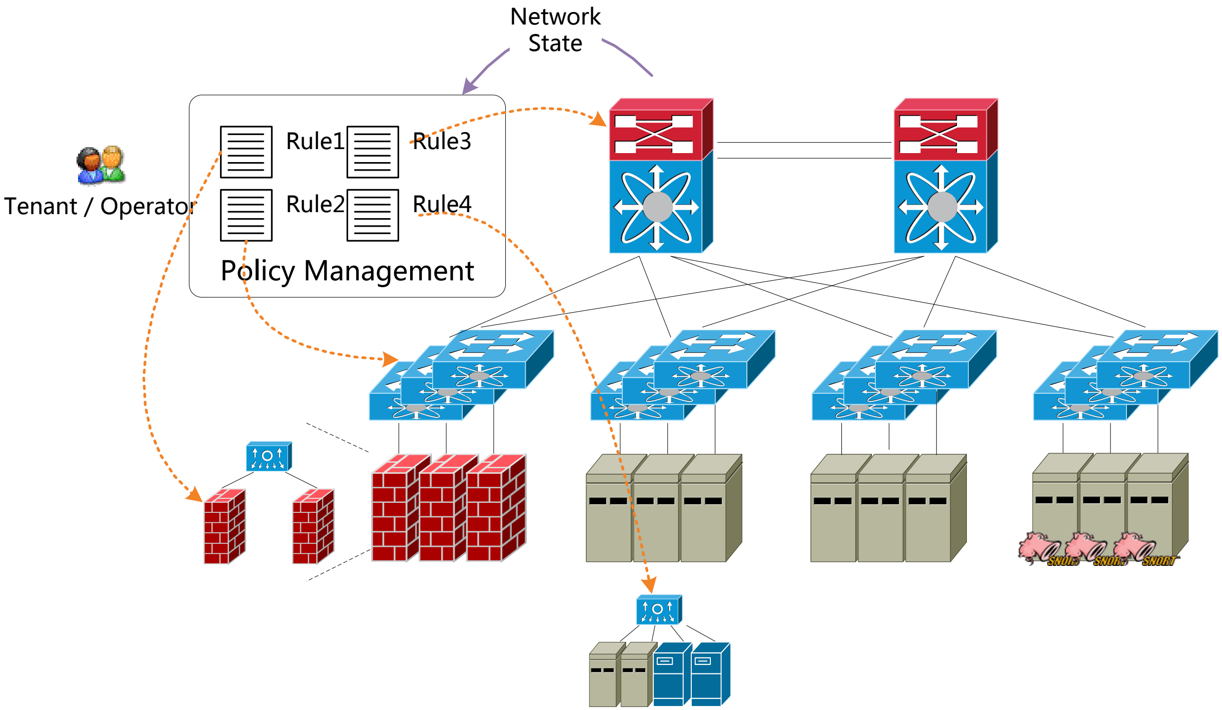
Tenant view



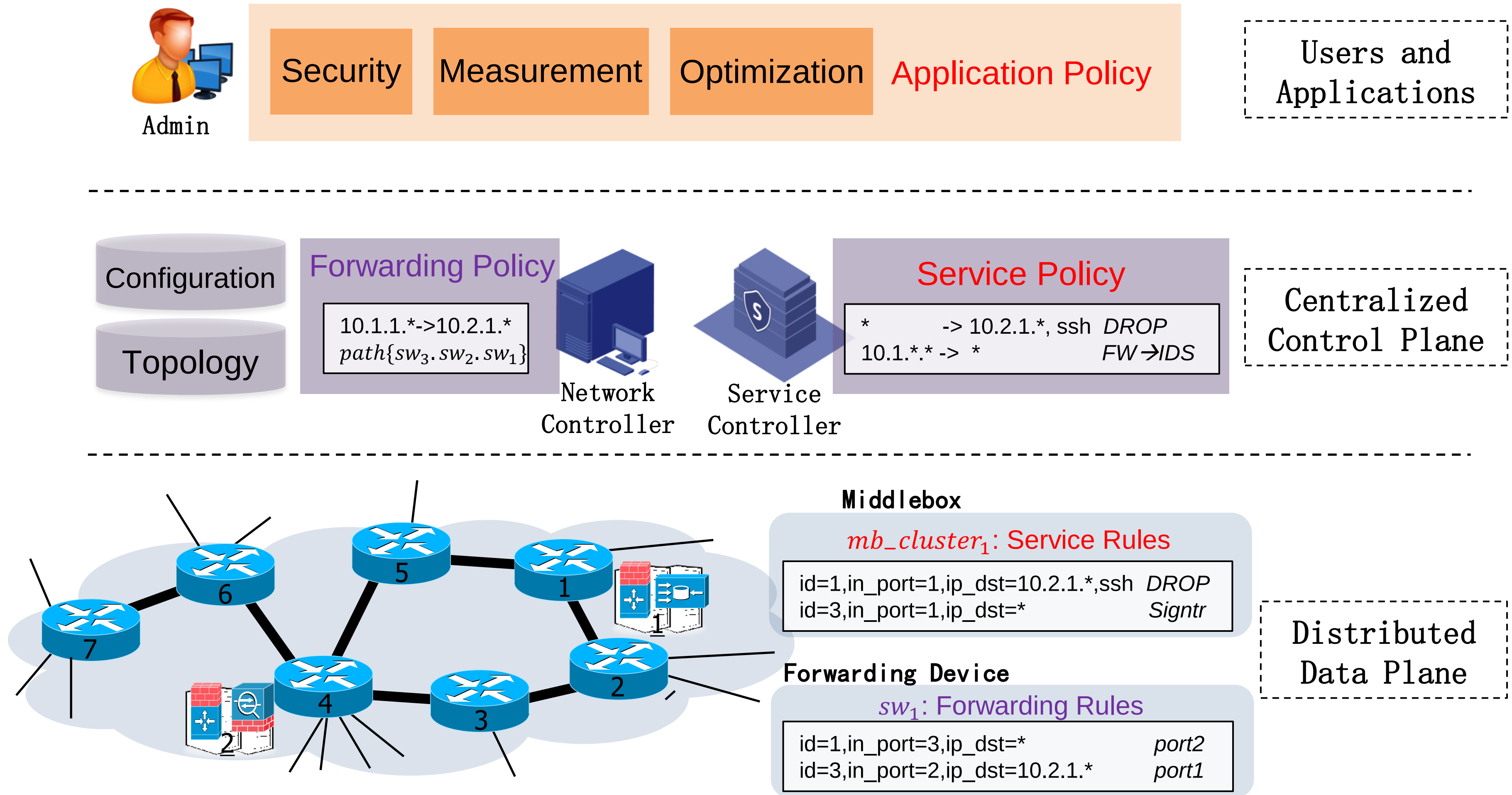
Operator view

Policy in Cloud DCN

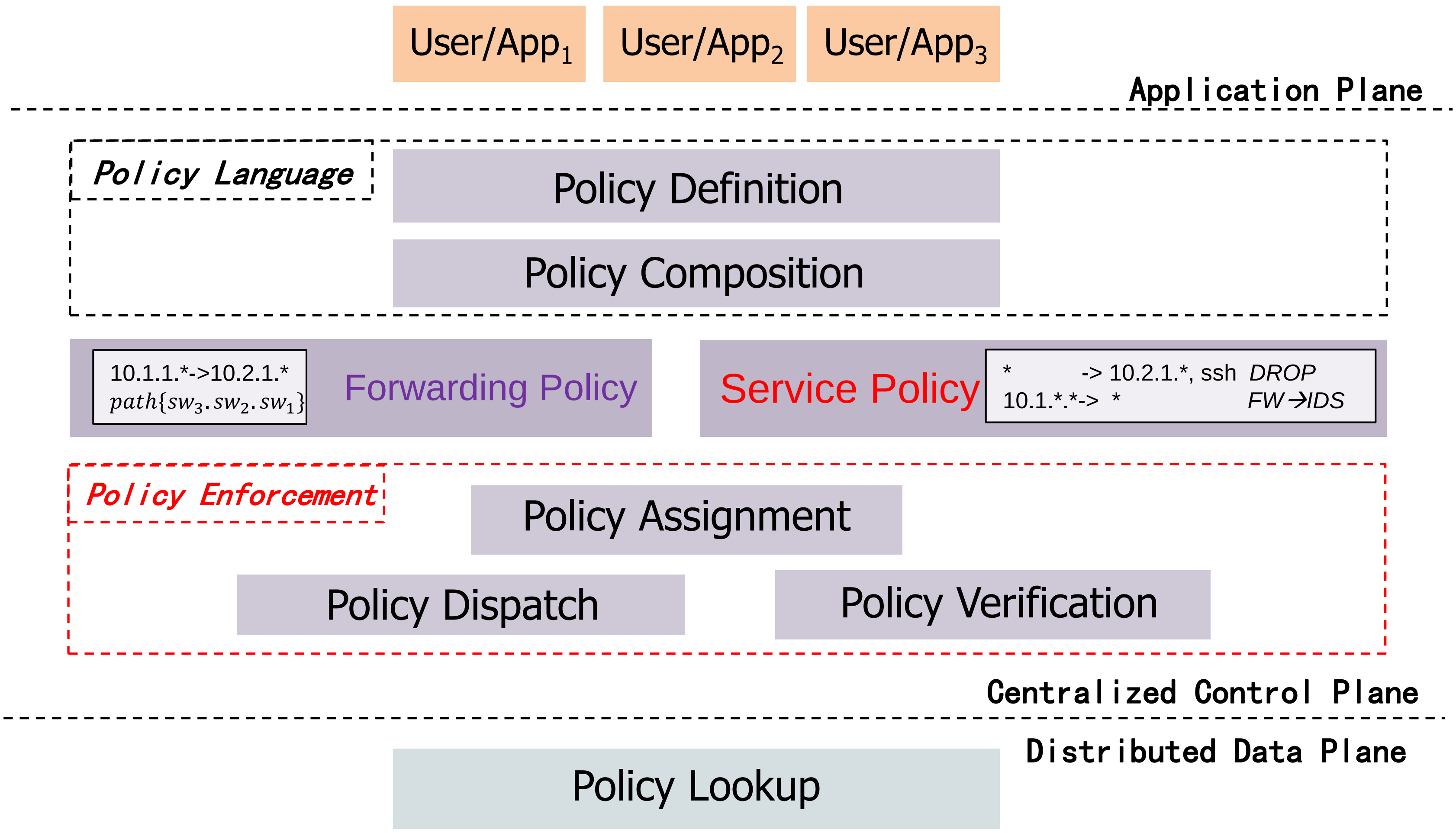
- Network Service Policy
  - Global vs. Local
  - Distributed and Dynamic Interaction



Policy Management (I)



Policy Management (II)



Outline

- Background
- Related Work
- Policy Space Analysis
- Policy Enforcement with PSA

Policy Language

- Policy Definition
  - DATALOG-based query
    - FML *[T. L. Hinrichs et al., WREN'09]*
  - Logical labels
    - PGA *[C. Prakash et al., SIGCOMM'15]*
- Policy Composition
  - High-level language for writing and composing modules
    - Frenetic *[N. Foster et al., SIGPLAN'11]*
    - Pyretic *[C. Monsanto et al., NSDI'13]*

Policy Enforcement (I)

- Policy Assignment
  - Distributed policies on switches w/ forwarding change
    - DIFANE *[M. Yu et al., SIGCOMM'11]*
    - vCRIB *[M. Moshref et al., NSDI'13]*
  - Distributed policies on switches w/o forwarding change
    - Palette *[Y. Kanizo et al., INFOCOM'13]*
    - One-Big-Switch *[N. Kang et al., CoNEXT'13]*
  - Distributed policies on middleboxes w/ forwarding change
    - SIMPLE *[Z. A. Qazi et al., SIGCOMM'13]*
  - Distributed policies on middleboxes w/o forwarding change
    - MBPE *[X. Wang et al., TON'16]*

Policy Enforcement (II)

- Policy Dispatch
  - Incremental updates
    - Update Abstractions *[M. Reitblatt et al., SIGCOMM'12]*
    - CCG *[W. Zhou et al., NSDI'15]*
  - Minimal flow-table
    - DevoFlow *[A. R. Curtis et al., SIGCOMM'11]*
- Policy Verification
  - Firewall policy analysis
    - FDD *[M. G. Gouda et al., ICDCS'04]*
  - Header space analysis
    - HSA *[P. Kazemian et al., NSDI'12 & 13]*
  - Real-time verification
    - Veriflow *[A. Khurshid et al., NSDI'13]*

Outline

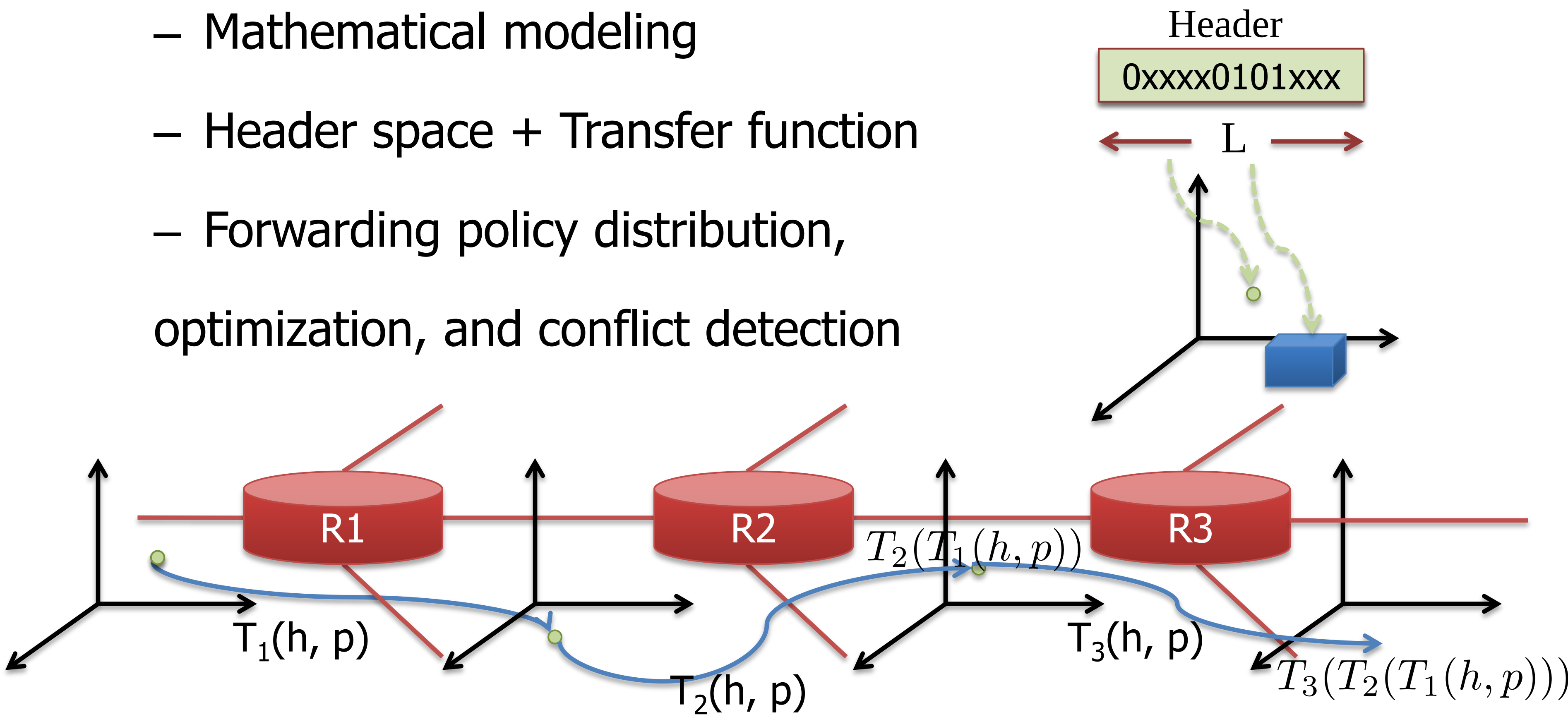
- Background
- Related Work
- Policy Space Analysis
  - Motivation
  - Design
  - Evaluation
- Policy Enforcement with PSA

Motivation

- What is the model and theoretical foundation of service policy?
- What is the common policy enforcement functionalities for assignment, dispatch, and verification?
- What is the performance requirements for practical policy enforcement?

Header Space Analysis (HSA)

- A simple abstraction to model all kinds of forwarding functionalities *[P. Kazemian et al., NSDI 2012 & 2013]*
  - Mathematical modeling
  - Header space + Transfer function
  - Forwarding policy distribution, optimization, and conflict detection



Problems with HSA

- HSA is inefficient for service policy management
  - Space representation in indiscriminate bits
    - The number of HSA computing dimensions is 104 for the classic 5-tuple policy.
    - Service policies usually contain arbitrary range values.
  - Set operations in an overlapping manner
    - Header space *(xxxx)* minus point *(1010)* is *(xxx1)* union *(xx0x)* union *(x1xx)* union *(0xxx)*, resulting in more computing tasks and duplicated sub-spaces while doing set operations.
  - Lack of efficient indexing data structures
    - Set operations are conducted in a linear manner.

Outline

- Background
- Related Work
- Policy Space Analysis
  - Motivation
  - Design
  - Evaluation
- Policy Enforcement with PSA

Policy Space Analysis

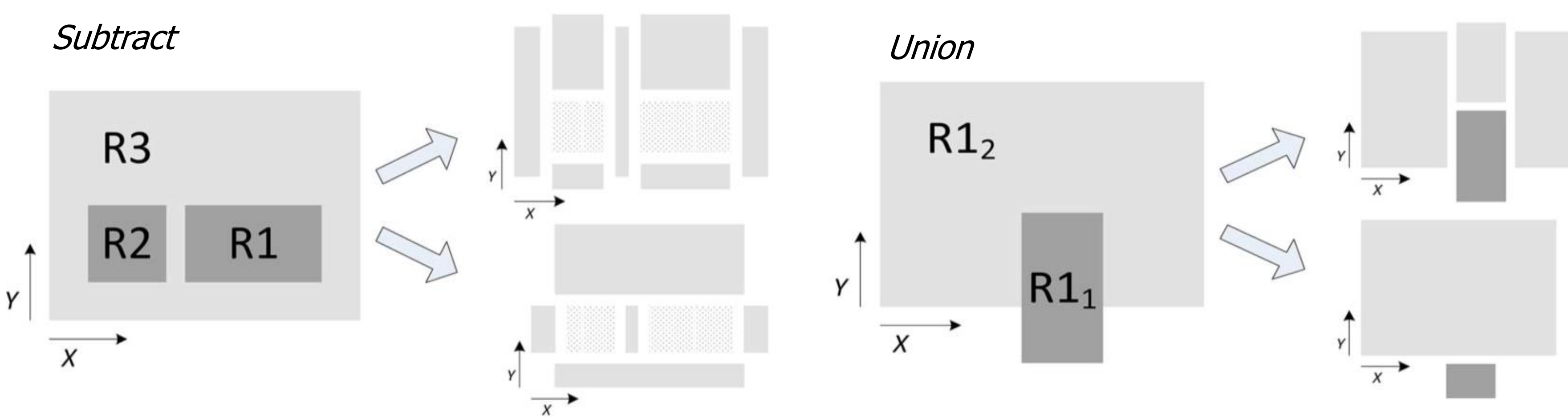
PSA

[X. Wang et al., TON 2016]

- Computational geometry view
  - Multi-dimensional space
- Expression
  - *HyperRect*: a D-field rule is viewed as a range-based D-dimension hyper-rectangle
  - *PolicySpace*: a set of multiple non-overlapping *HyperRects*
- *Boolean and Set Operations*
  - Supported by both *HyperRect* and *PolicySpace*
  - *Boolean Operations*
    - *is_equal*, *is_subset*, and *is_intersected*
  - *Set Operations*
    - *intersect*, *subtract*, and *union*

PSA Implementation

- *HyperRect Operation*
  - Different dimension inspecting sequences produce diverse operation results
- *PolicySpace Operation*
  - Most operations implemented as the iteration of the same operations of the *HyperRect*
  - *is_equal* implemented as bidirectional *is_subset* operations
  - *is_subset* implemented as volume comparison of the minuend policy space and the intersected policy space based on the non-overlapping property

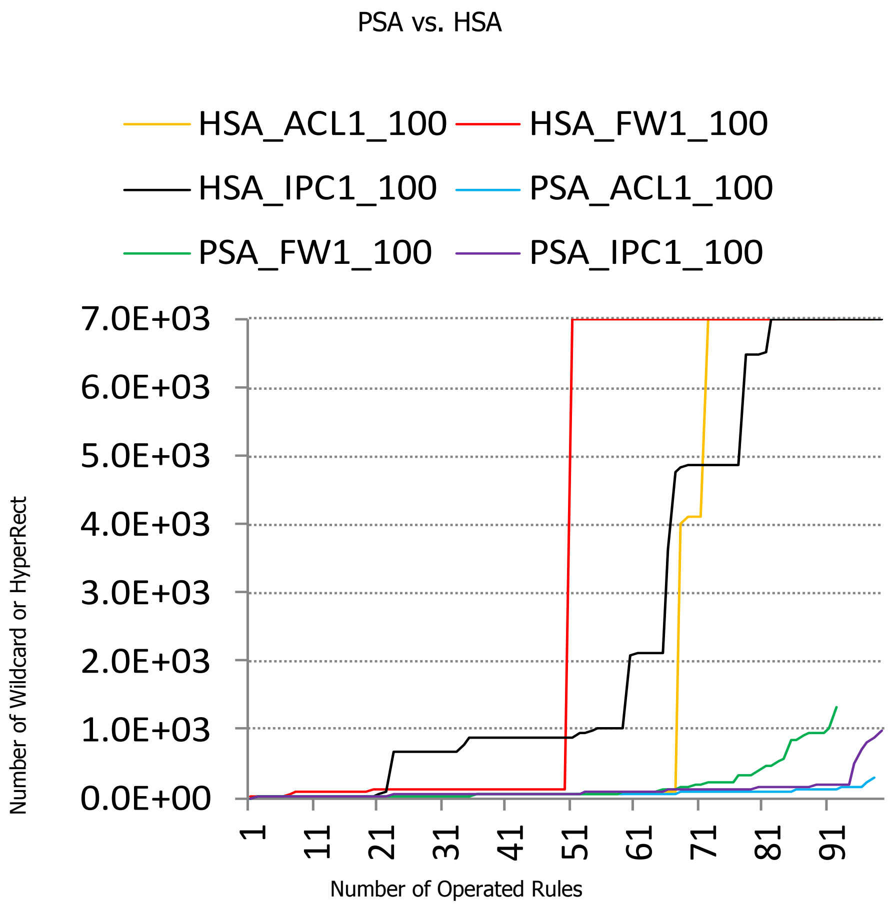


Outline

- Background
- Related Work
- Policy Space Analysis
  - Motivation
  - Design
  - Evaluation
- Policy Enforcement with PSA

PSA Evaluation (I)

- Spatial performance
  - Iterated *subtraction* of high-priority rules from low-priority rules to construct a non-overlapping ruleset

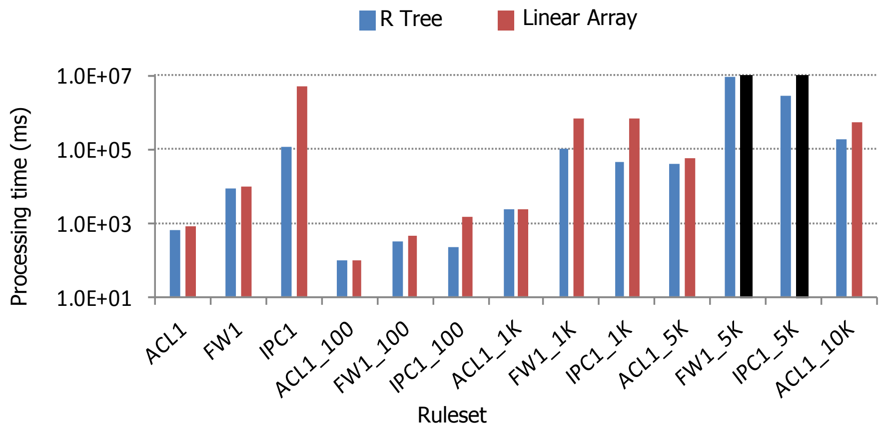


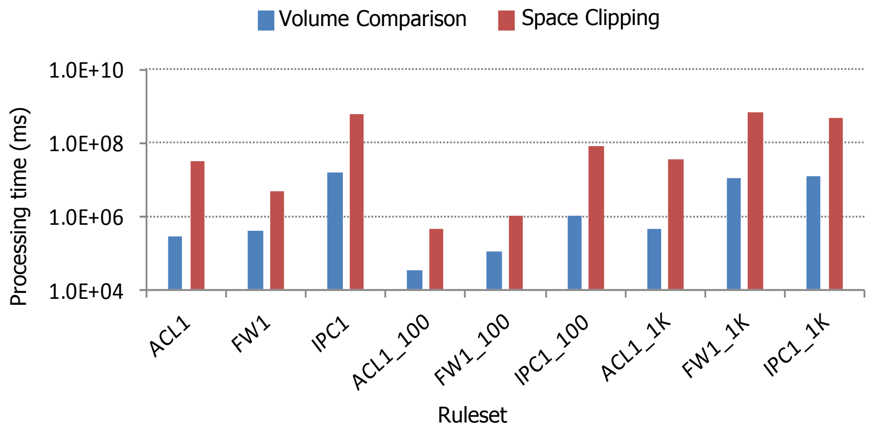
PSA Evaluation (II)

- Temporal performance

— Iterated *subtraction* of high-priority rules from low-priority rules to construct a non-overlapping ruleset

— *is_equal* operation between the union of non-overlapping rules and the union of overlapping rules





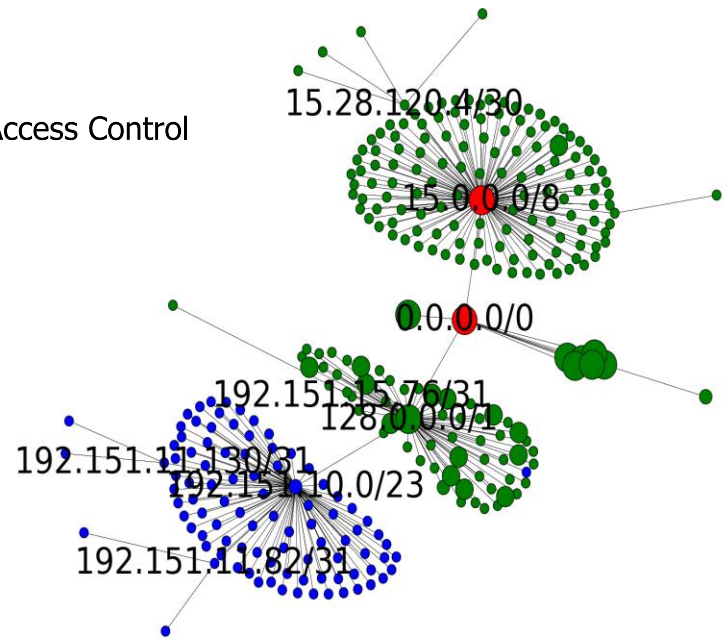
Outline

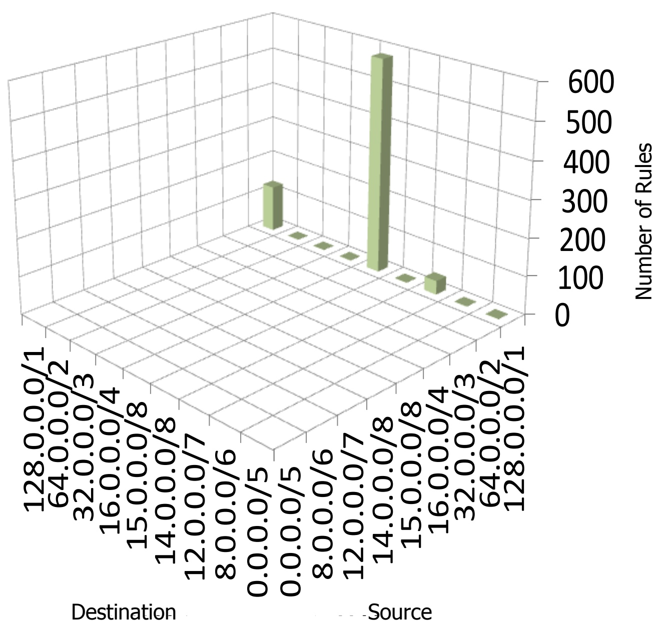
- Background
- Related Work
- Policy Space Analysis
- Policy Enforcement with PSA
  - Topological Analysis of Service Policy
  - Policy Assignment
  - Policy Verification

Topological Analysis of Service Policy (I)

- Analysis of three classical policies
  - Policy topology: Hierarchical addresses
  - Topological statistics: Rule distribution

Access Control

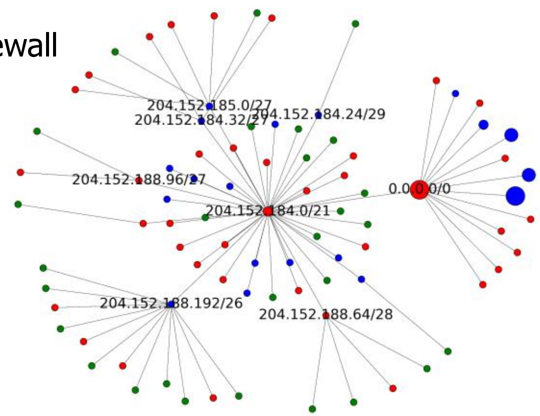


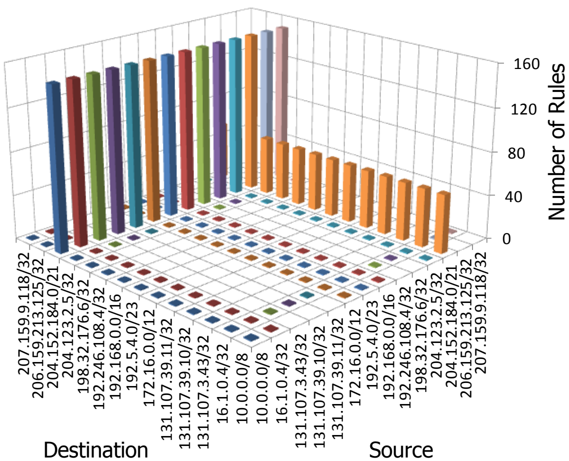


(exact rules)

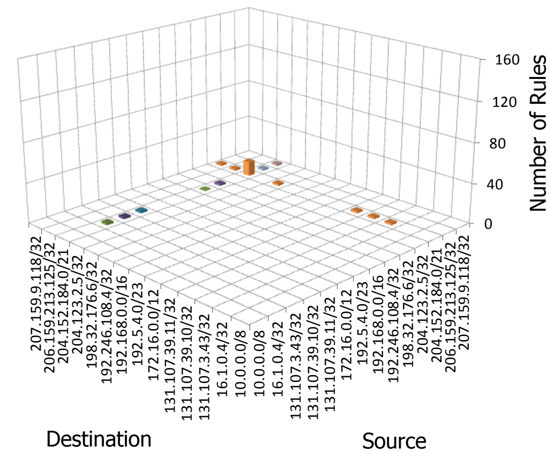
Topological Analysis of Service Policy (II)

Firewall



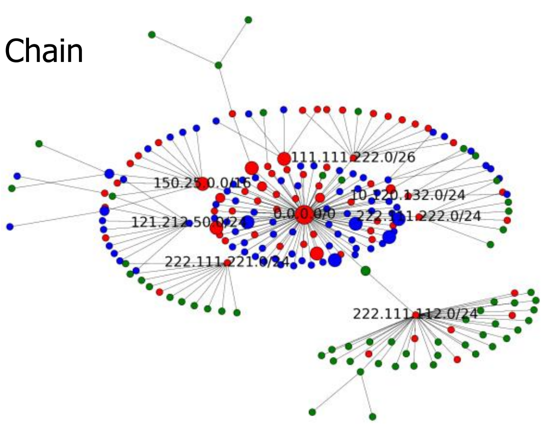


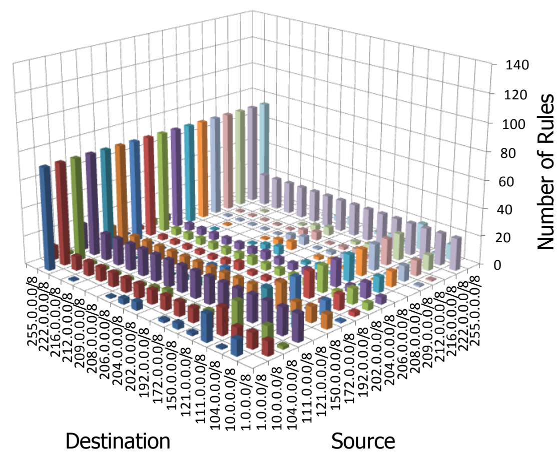
(partially specified rules)



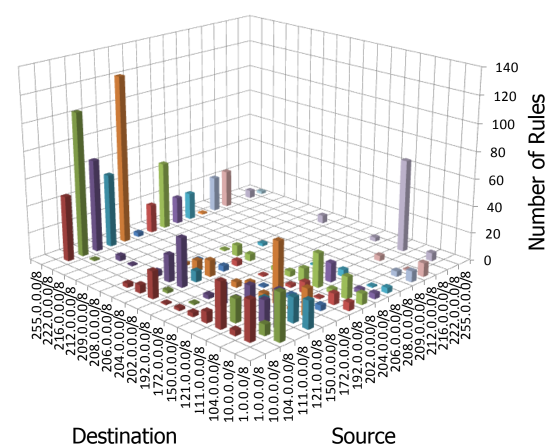
(exact rules)

IP Chain





(partially specified rules)



(exact rules)

Topological Analysis of Service Policy (III)

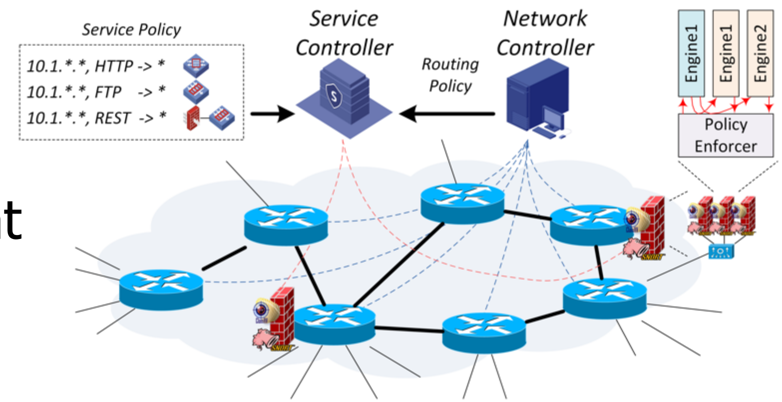
- Forwarding policy vs. Service Policy
  - Forwarding Policy
    - Entire network view, fine-grained address objects, less rule overlapping
  - Service Policy
    - Choke points of subnets, relationship of organization, wildcard and more rule overlapping
- Implication for packet classification
  - Rule distribution is asymmetrical
  - Firewall and IP Chain type scenarios have more overlapping
  - Either source IP or destination IP is highly separable

Outline

- Background
- Related Work
- Policy Space Analysis
- Policy Enforcement with PSA
  - Topological Analysis of Service Policy
  - Policy Assignment
  - Policy Verification

MBPE

MiddleBox Policy Enforcement



- Principles
  - Preserve Forwarding Rules
    - On-datapath processing to reduce bandwidth cost introduced by traffic steering
  - Dispatch onto middleboxes
    - From path-wise to network-wise to reduce wildcard rules replication
- Requirements
  - Correctness
    - Each policy or policy partition is processed once and only once
  - Efficiency
    - As less enforced nodes as possible

Set Cover Problem (SCP)

- Given a set ***U*** of ***n*** elements and a collection ***S*** of subsets of ***U***, find the smallest collection ***C*** of ***S*** whose union is ***U***
- $\boldsymbol{S_P}$ is mapped to ***U*** and $\boldsymbol{S_f}$ is mapped to ***S***

MBPE modeled as SCP

$$min \sum_{j=1}^{N} x_j$$

subject to

$$\forall i \in \{1, \dots, M\}: \bigcup_{j=1}^{N} S_{r_i}^{j} = S_{r_i}, S_{r_i}^{j} \subseteq S_{r_i} \cap S_f^{j}$$

$$\forall j_1, j_2 \in \{1, \dots, N\}, j_1 \neq j_2, \forall i \in \{1, \dots, M\}: S_{r_i}^{j_1} \cap S_{r_i}^{j_2} = \emptyset$$

$$x_j = \begin{cases} 1, & \exists i \in \{1, \dots, M\}, s.t. S_{r_i}^{j} \neq \emptyset \\ 0, & otherwise \end{cases}$$

S_f	The *flow space* of all network nodes
S_f^j	The *flow space* that the traffic covers at enforced node *j*
S_{r_i}	The *rule space* that i^{th} rule covers
$S_{r_i}^j$	The *rule space* that i^{th} rule covers at enforced node *j*

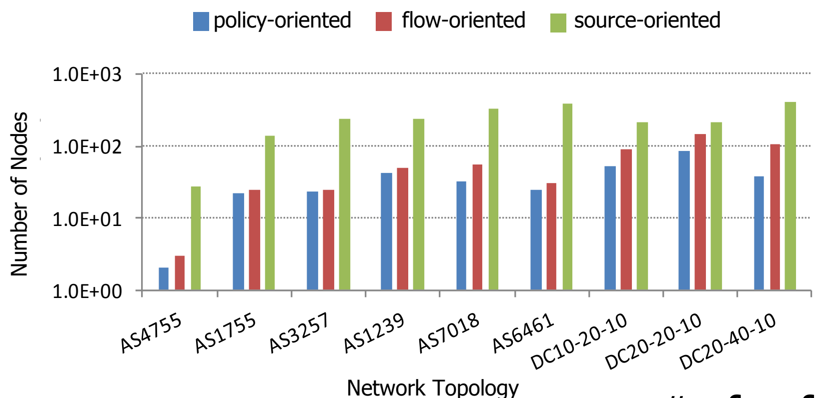
Greedy Algorithm

- Iteratively select the network node whose *flow space* can cover most of the remaining *policy space*
- Enforce two types of rules in each iteration
  - Bypass rule with high priority: subtraction of the remaining flow space from the flow space of the selected node
  - Enforced rule with low priority: intersection of the remaining flow space of the selected node and the complete policy space
- Two heuristics to select network nodes
  - Policy-oriented: node intersects with the maximum number of policy rules
  - Flow-oriented: the node has the maximum number of hyper-rectangles

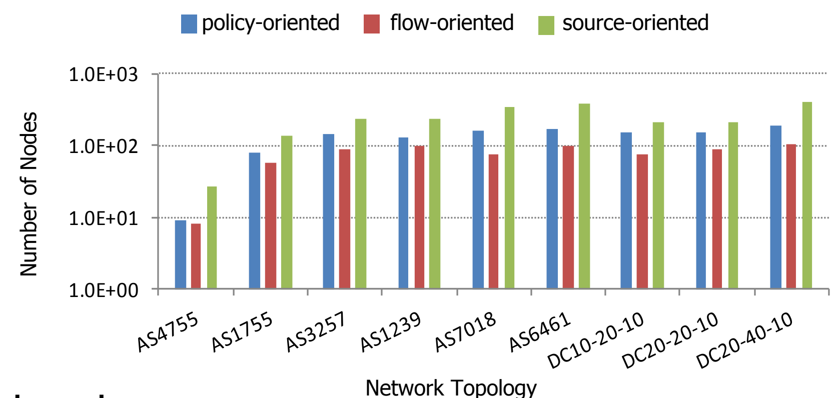
Evaluation

- Enforced nodes and rules

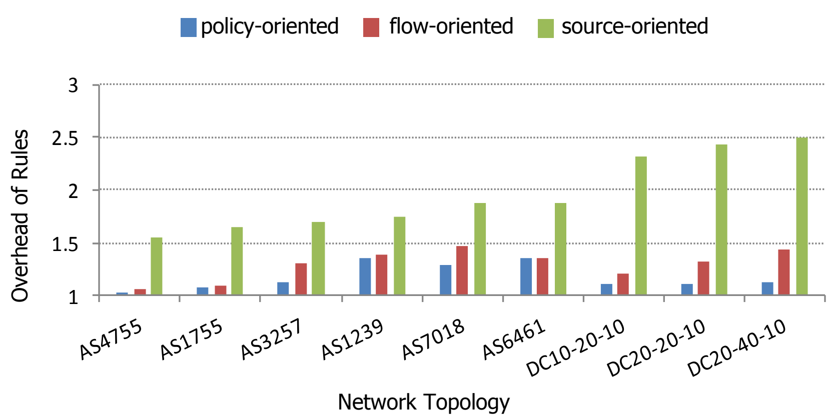
exact policies

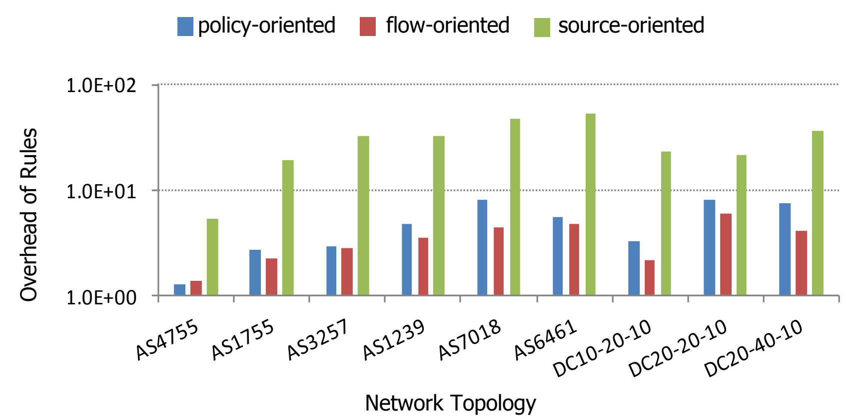


wildcard policies



# of enforced nodes





# of all enforced rules / # of the original policy rules

Extended MBPE

- Practical constraints to Set Cover Problem
  - Rule size of forwarding devices
  - Processing capability of middleboxes

$$min \sum_{j=1}^{N} x_j$$

subject to

$$\forall i \in \{1, \dots, M\}: \bigcup_{j=1}^{N} S_{r_i}^j = S_{r_i}, S_{r_i}^j \subseteq S_{r_i} \cap S_f^j$$

$$\forall j_1, j_2 \in \{1, \dots, N\}, j_1 \neq j_2, \forall i \in \{1, \dots, M\}: S_{r_i}^{j_1} \cap S_{r_i}^{j_2} = \emptyset$$

$$\forall j \in \{1, \dots, N\}: \sum_{i=1}^{M} F(S_{r_i}^j) \leq C_j$$

$$x_j = \begin{cases} 1, & \exists i \in \{1, \dots, M\}, s.t. S_{r_i}^j \neq \emptyset \\ 0, & otherwise \end{cases}$$

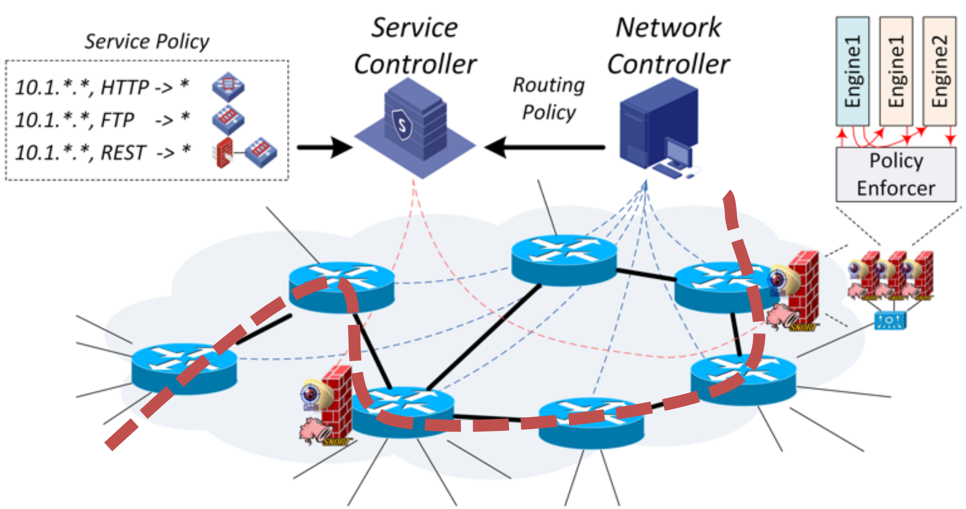
S_f	The *flow space* of all network nodes
S_f^j	The *flow space* that the traffic covers at enforced node j
S_{r_i}	The *rule space* that i^{th} rule covers
$S_{r_i}^j$	The *rule space* that i^{th} rule covers at enforced node j
$F()$	Cost mapping function
C_j	Constraints of node j

Outline

- Background
- Related Work
- Policy Space Analysis
- Policy Enforcement with PSA
  - Topological Analysis of Service Policy
  - Policy Assignment
  - Policy Verification

Policy Verification

- Distributed data plane policies verification
  - Decompose network scope problem into path scope problems according to forwarding policy
  - Operate policies along the specified flow path
  - Leverage the set operations of PSA



Policy Verification

- Distributed data plane policies verification
  - Consistency
    - $\overline{S_{f'}} = S_{f'} \cap S_P$
  - Redundancy
    - $\exists\, j_1 \neq j_2 : \overline{S_{f'}^{j_1}} \subseteq \overline{S_{f'}^{j_2}}, \overline{S_{f'}^{j_1}}.action = \overline{S_{f'}^{j_2}}.action$
  - Confliction
    - $\exists\, j_1 \neq j_2 : \overline{S_{f'}^{j_1}} \cap \overline{S_{f'}^{j_2}} \neq \emptyset, \overline{S_{f'}^{j_1}}.action \neq \overline{S_{f'}^{j_2}}.action$

f'	flow of one network policy
$S_{f'}$	flow space
S_P	The *policy space* of all *rule space*
$\overline{S_{f'}^{j}} = \bigcup_{i=1}^{M} S_{r_i}^{j} \cap S_{f'}$	rule space on node j
$\overline{S_{f'}} = \bigcap_{t=1}^{K} \overline{S_{f'}^{j_t}}$	combined rule space on $path$ of flow f'

Summary

- An orthogonal perspective of network forwarding and network service
- A framework of policy space analysis definitions and operations
- A few policy enforcement algorithms as research in progress

Thanks