Acila:
Attaching Identities of Workloads for Efficient Packet Classification in a Cloud Data Center Network

Kentaro Ohnishi, Daisuke Kotani (Kyoto University)
Hirofumi Ichihara, Yohei Kanemaru (LINE)
Yasuo Okabe (Kyoto University)

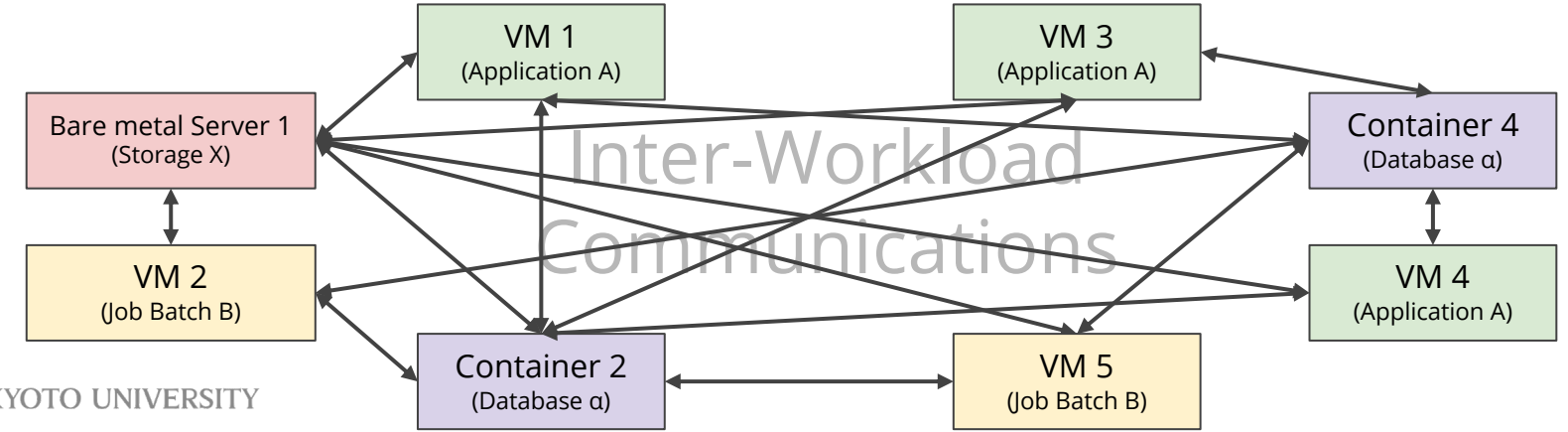
京都大学

Network Control in Cloud

Various software (workload) are running in different styles on cloud
Virtual machines, Containers or Pods, Bare metal servers, etc

Enforcing some control on packets between workloads is essential

- Packet filtering to maintain security
- QoS to keep application performance



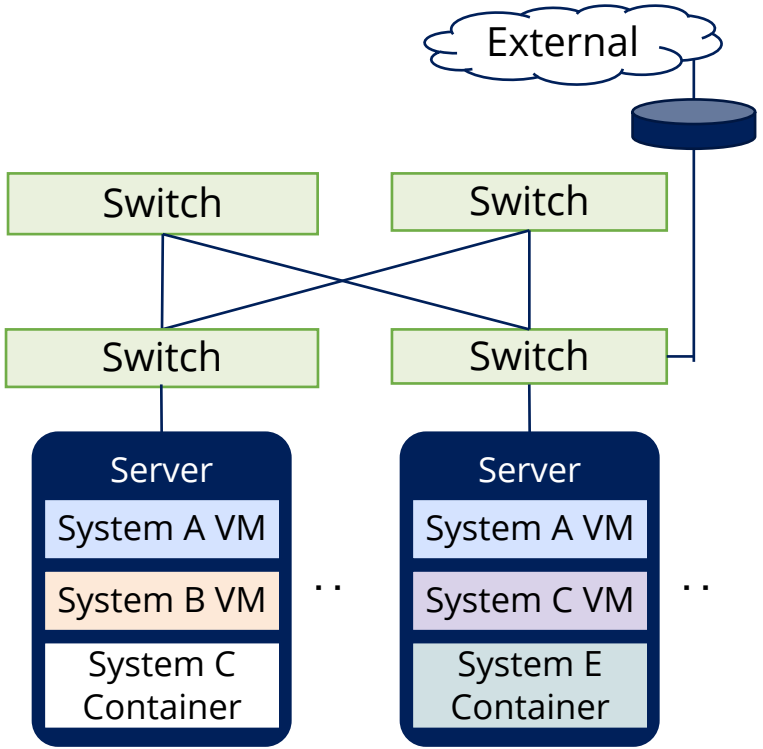
Cloud Infrastructure

Clos network topology + IP routing

VMs, containers, etc. (workloads) are elastically deployed on servers where resources are available

- Workloads of multiple systems coexist in one server (hypervisor)
- IP addresses are assigned based on locations

External connections to other networks (legacy systems, enterprise networks, etc)



Rules for Network Control

Traditional Way:
Workloads of each system has
IP addresses from the same IP prefix
=> use IP prefix (+ port) for rules

From \ To	System A 192.0.2.0/26	System B 192.0.2.64/26
System A	Allow	Deny
System B	Allow	Allow

192.0.2.0/26, *, Allow
192.0.2.64/26, *, Deny

In Cloud:
Workloads has IP addresses assigned
from the IP prefix allocated to the host
=> use IP address (+ port) for Rules

From \ To	System A 192.0.2.1 192.0.2.3	System B 192.0.2.2 192.0.2.4
System A	Allow	Deny
System B	Allow	Allow

192.0.2.1, 192.0.2.2, Deny
192.0.2.1, 192.0.2.4, Deny

# of entries and their update frequency will be explosively increased

Related Work

Attach a tag to packets to deliver what the source is (Cilium, eZTrust)

- Policies requiring both src and dest info can be applied **only at the dest** because packets lack the destination info

Deploy overlay networks and assign IP addresses in the traditional way

- Management overhead (incl. interconnection between networks)
  - Especially for microservices where a system consists of multiple loosely-coupled services

Separation of host identifiers and locators (HIP, Mobile IP, LISP)

- Host level granularity are not necessary for the rules

Our Work

Q: Can we design and use identifiers for implementing variety of network control on cloud in a simple and efficient way?

- Supporting VMs, containers, legacy systems via external connections
- Network controls can be applied at both ends and in-network

Our proposal: Add identifiers to packets showing what a client is and a server is (i.e. identity), by Acila

- Design how to generate identifiers and attach them to packets
- Implemetation using eBPF as data plane
- Applications and evaluation

Workload, Identity of Workload and Labels

Workload: a single piece of software, deployed with a particular configuration for a single purpose (by SPIFFE)

Deployed as VM, container, process in a VM, etc.

Identity of Workload: information that characterizes the workload

Represented as a set of pairs of key and value (called Labels)

Cloud controller (or orchestrator) has a database of identity of all workloads in the cloud.

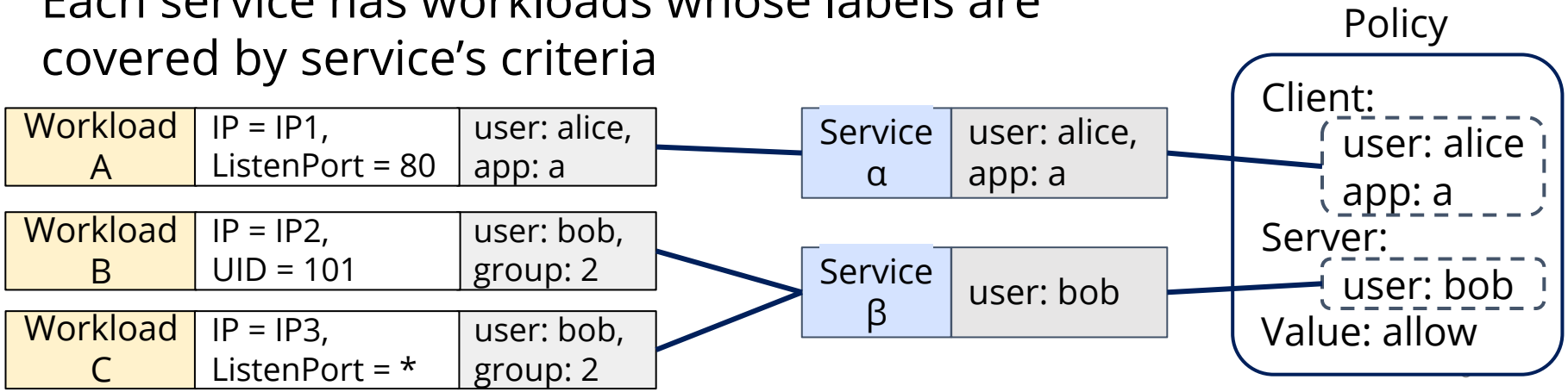
	Condition for Identification	Labels
Workload A (VM)	IP = IP1, ListenPort = 80	user: alice, app: a
Workload B (Process)	IP = IP2, UID = 101	user: bob, group: 2
Workload C (Container)	IP = IP3, ListenPort = *	user: bob, group: 2

Workload, Policy and Service

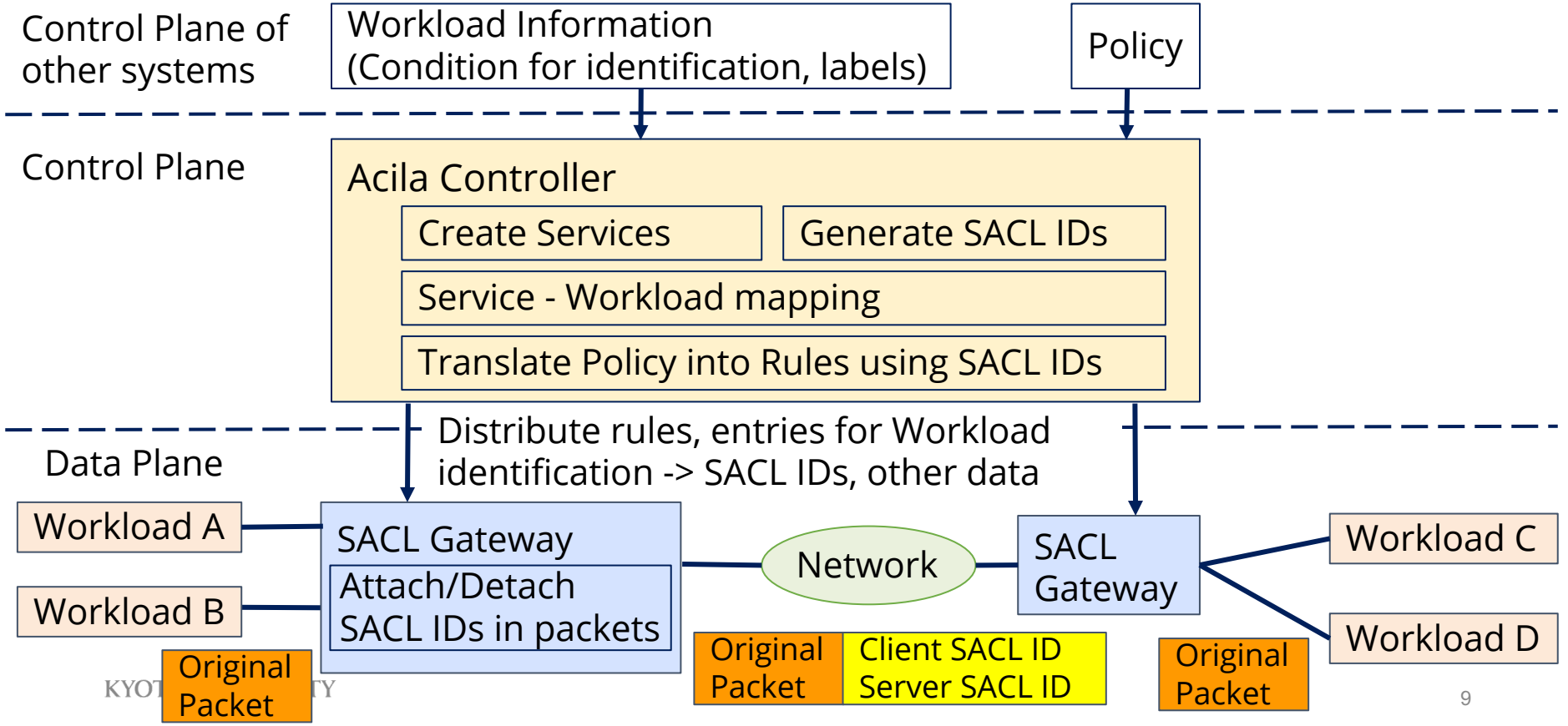
Service represents a group of workloads whose packets are processed in the same way, computed from Policy

- Use a subset of labels for describing workload selection criteria
- **SACL ID** is assigned to each service

Each service has workloads whose labels are covered by service's criteria



Acila at a Glance



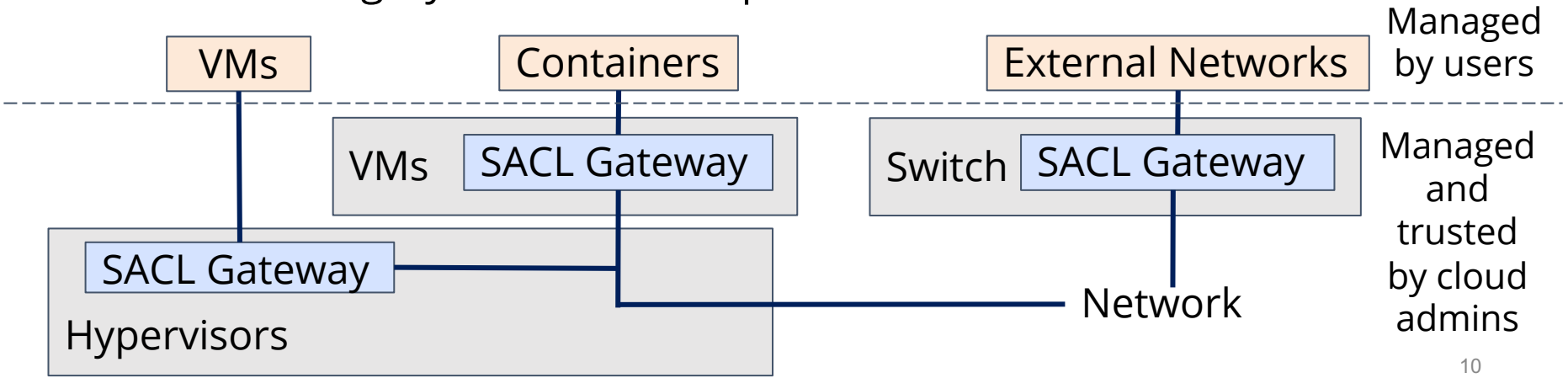
Data Plane - Where to Install SACL Gateway

Close to the workloads as much as possible where trusted by cloud admins

- Rich information is available to identify the workload

  Interface, IP address before translated (e.g. Virtual IP), etc

- Avoid forgery of SACL IDs in packets



Data Plane - Client Side and Server Side

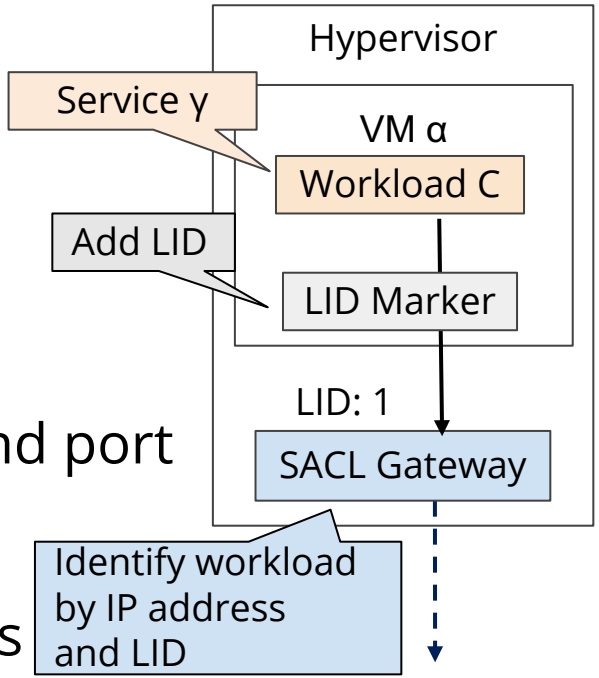
Translate Workloads into SACL IDs, and attach them to packets as IPv6 Hop-by-hop Option designed for Acila

SACL Gateway for Client Workload

- Identify Client Workload
  - VMs and Containers by IP address
  - Process by LID Marker inside the VM
- Identify Server Workload by IP address and port

SACL Gateway for Server Workload

- Use Conntrack to identify client workloads



Policy

Each policy consists of:

- Client Selectors: a set of (key, operator, values)
- Server Selectors: a set of (key, operator, values)
- Value: (optional, used for applications)

key: a key of labels of workloads to be searched for
operator: whether any value in “values” should exist (or not)
values: a set of values that should or should not exist

Acila Controller

SACL ID generation

- Currently labels of workloads and SACL ID have one-to-one mapping (just for simplicity, should improve in the future)

Workload identification and SACL ID mapping rule

- IP address and listen port (+ process UID via LID Marker) => SACL ID

Distribute rules for SACL Gateways and switches

Acila Applications

Packet Filtering:

- Use Client/Server SACL IDs for matching condition
- Can filter out packets anywhere in the network

Priority Packet Scheduling in The Network

- Use Client/Server SACL IDs for matching condition
- Deploy rules at all switches in the network

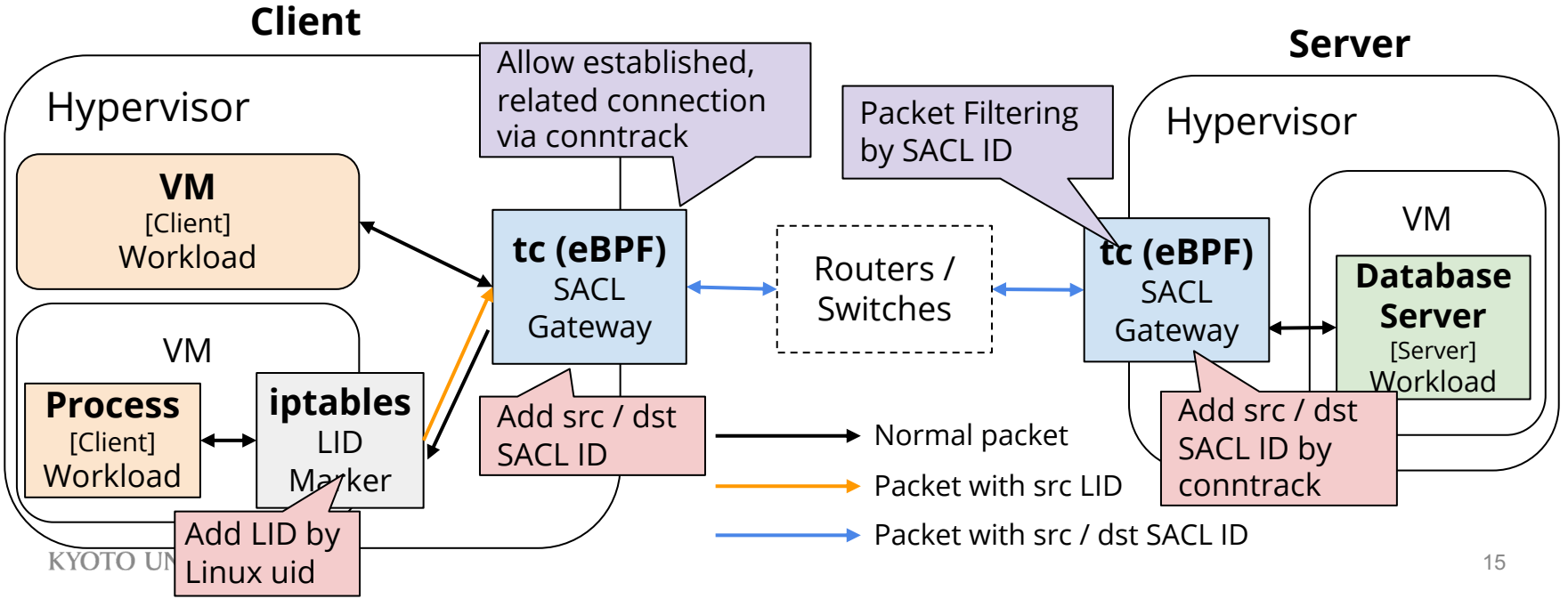
Acila is effective because there is no need to update rules according to elastic changes of workloads

c.f. IP address based rules should be updated once a workload is created or deleted

Implementation for Packet Filtering

Control Plane: Policies are interpreted as allowlist

Data Plane



Evaluation: Estimated # of Entries

Spine switches under priority packet scheduling in a clos network

Entries		Update Frequency by Workload Creation and Deletion	
IP address based	Acila	IP address based	Acila
$\sum_{w_i \in W} \sum_{\substack{s_k \in SS_{s_j} \\ (W_{s_j} \ni w_i)}} \lvert W_{s_k} \rvert$	$\sum_{s_i \in S} \lvert SS_{s_i} \rvert$	$\sum_{s_k \in SS_{s_j}} \lvert W_{s_k} \rvert + \sum_{s_l \in CS_{s_j}} \lvert W_{s_l} \rvert \quad (W_{s_j} \ni w_i)$	0

W: All Workloads, W_{sj}: Workloads belonging to Service s_j, S: All Services
SS_{sj}: Server Services accessed by Client Service s_j with priority
CS_{sj}: Client Services that access Server Service s_j with priority

Clearly, Acila requires less entries and less entry updates

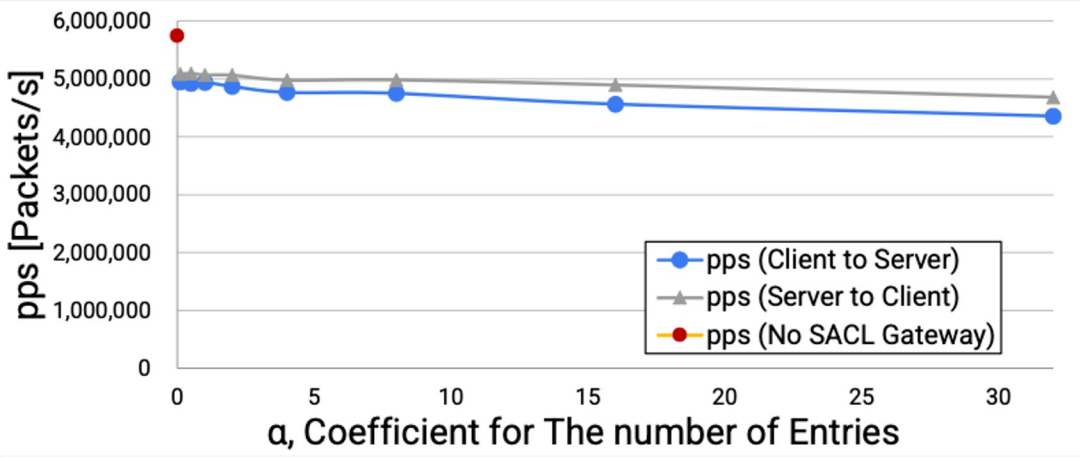
More on our paper and preprint (arxiv:2109.08343)

Evaluation: Performance of SACL Gateway

SACL Gateway will have many entries for mapping IP addresses, ports, and LIDs to Client/Server SACL IDs as well as Conntrack

Measure impact on packet forwarding performance by entries

Assumption of entries about # of VMs, Workloads, Connections between Services:
$1.3 \times 10^2 \alpha$ for Client SACL IDs,
$3.8 \times 10^3 \alpha$ for Server SACL IDs,
$3.8 \times 10^3 \alpha$ for Conntrack



Environment:
Two machines connected by 100GbE, AMD EPYC 7282, 128GB RAM

Concluding Remarks and Future Work

In cloud, IP address is useless for identifying endpoint in network control

Acila provides a new identifiers, SACL IDs, attached to packets for identifying endpoints in network control

SACL IDs are generated from identity of workloads and policies

SACL IDs are useful for various applications

at least for packet filtering and priority packet scheduling

Future work includes

- Improve Workload identification and SACL IDs generation process
- Support/implement for other cases, such as legacy systems and NIC offload