SDNSOC: Object Oriented SDN Framework

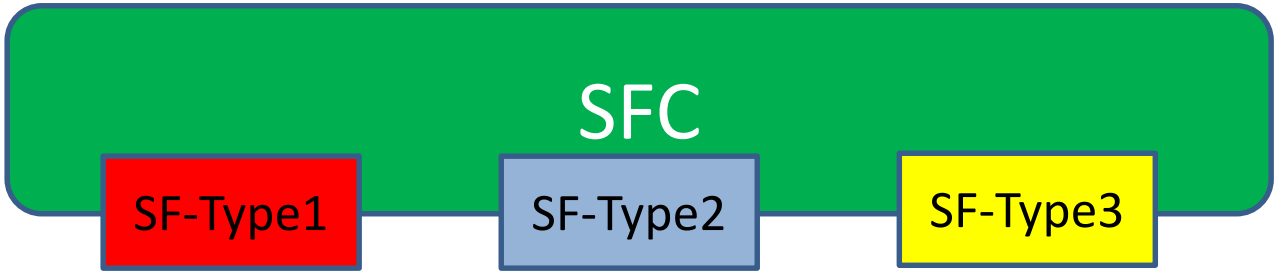
Ankur Chowdhary, Dijiang Huang, Gail-Joon Ahn,
Myong Kang, Anya Kim and Alexander Velazquez
Arizona State University
US Naval Research Lab
SDNNFVSec, March 27 2019

Ira A. Fulton Schools of Engineering
ARIZONA STATE UNIVERSITY

Agenda

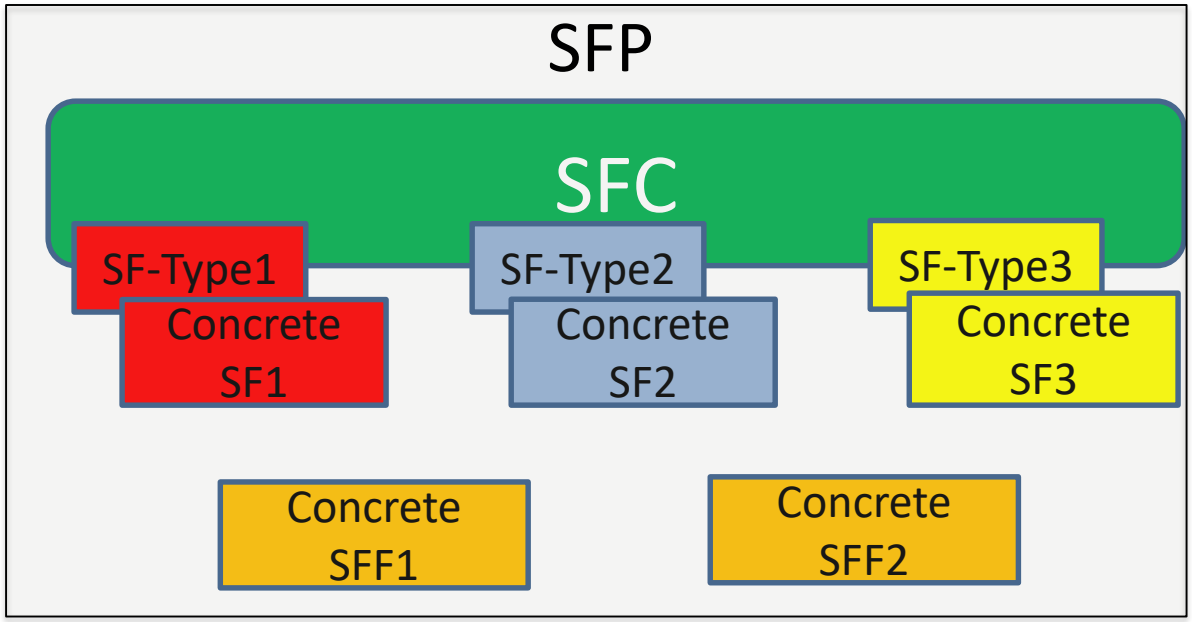
1. **Introduction**
   - Policy Composition Challenged
   - OpenFlow rule conflicts
   - Need for programmatic framework
2. Motivation and Background
3. SDNSOC Architecture
4. Object Oriented Framework
5. Implementation and Evaluation
6. Conclusion and Future Work

Service Function Chaining



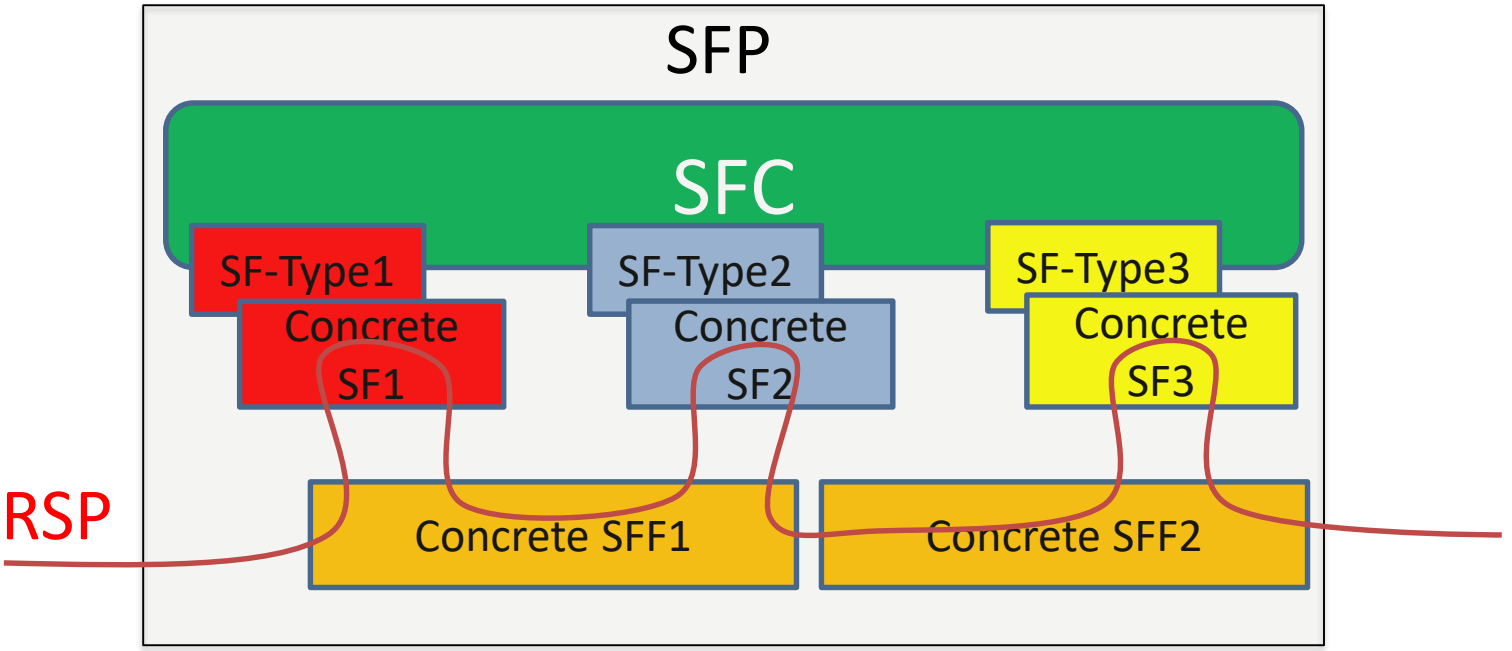
- Abstract Ordered List of Service Function types.
- Service Function: Function responsible for specific treatment of received packets.
- E.g. – DPI, FW, NAT

Service Function Path (SFP)



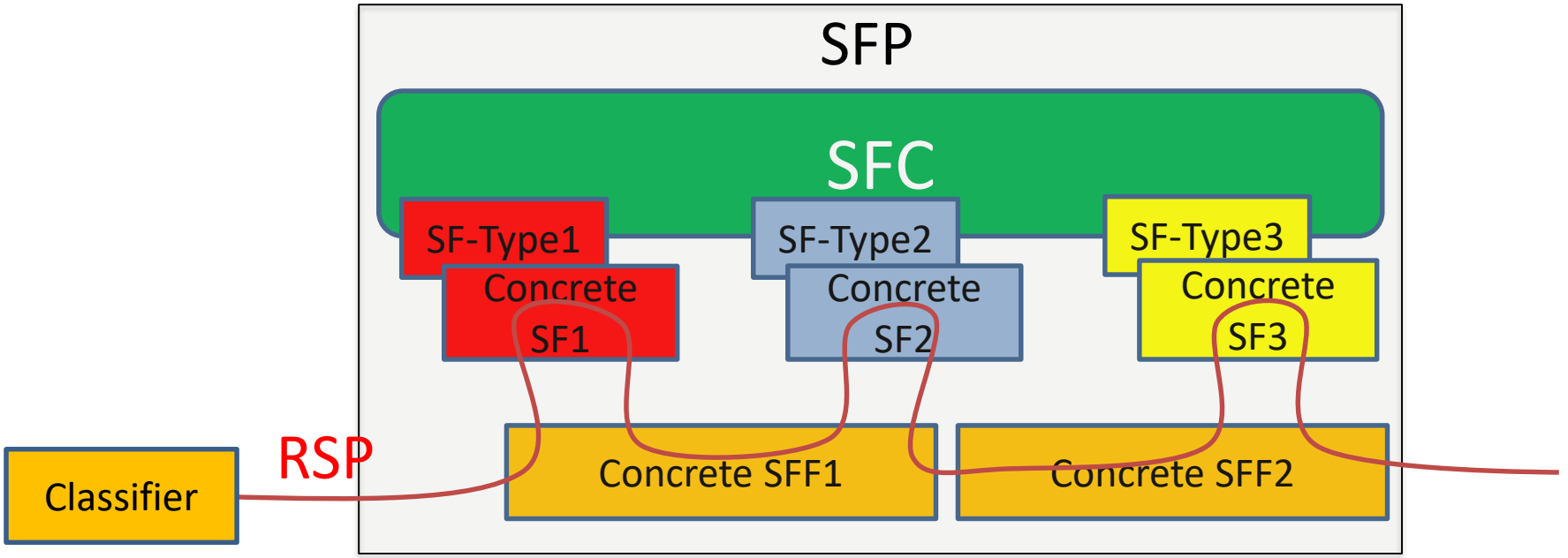
- Provides directional details about SFC.
- Underlay and overlay transport details (VxLAN-NSH, Eth-NSH).
- Concrete SFs and Service Function Forwarders (SFFs).

Rendered Service Path (RSP)



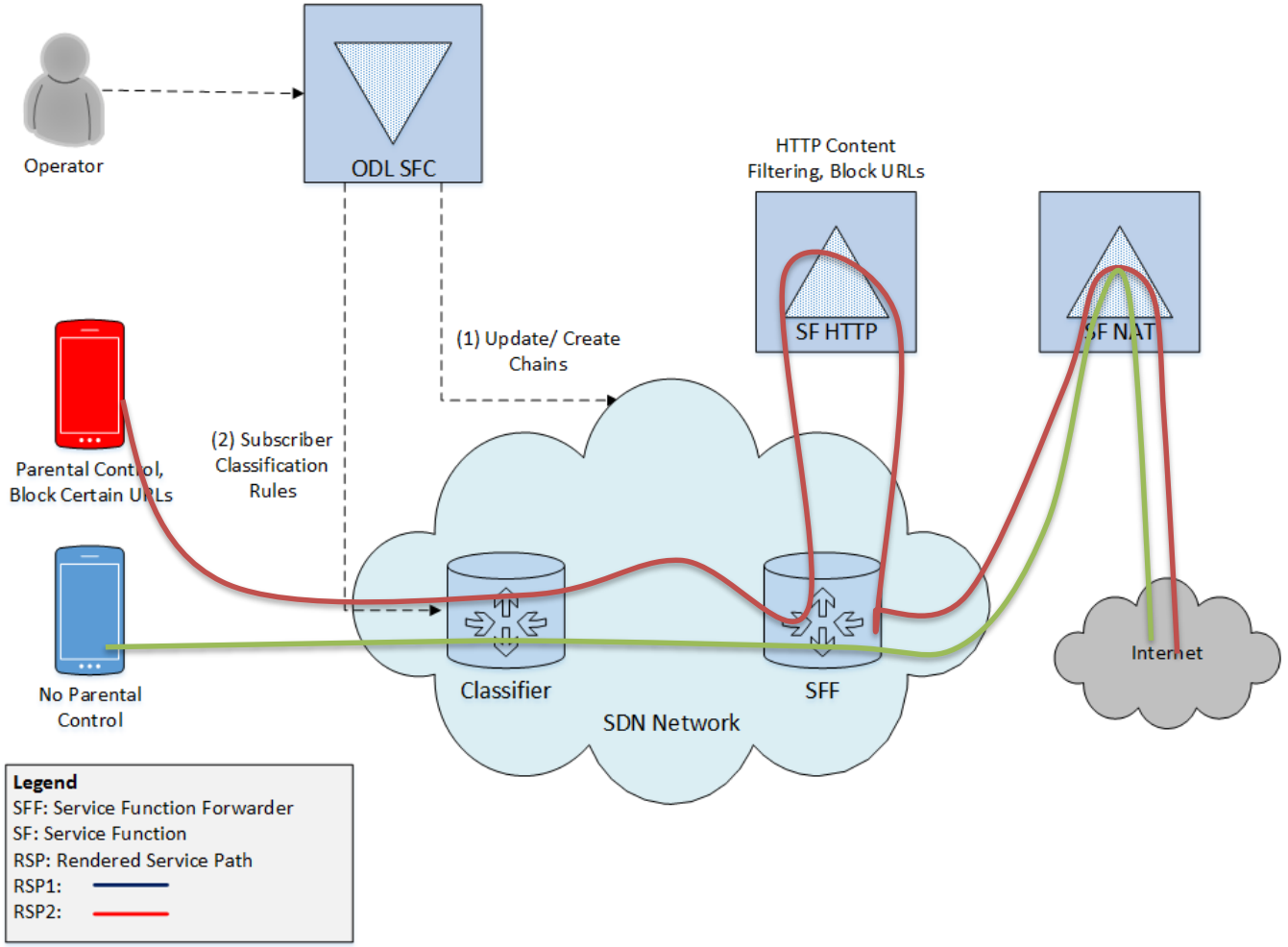
- Actual service chain combining SFC and SFP
- Runtime representation of SFC.

Service Chain Classification



- Traffic flow matching based on Access Control List (ACL).
- Subscriber-tenant traffic flows to Service Chain mapping.
- Service Chain Encapsulation (NSH).

Service Function Chain : Use Case



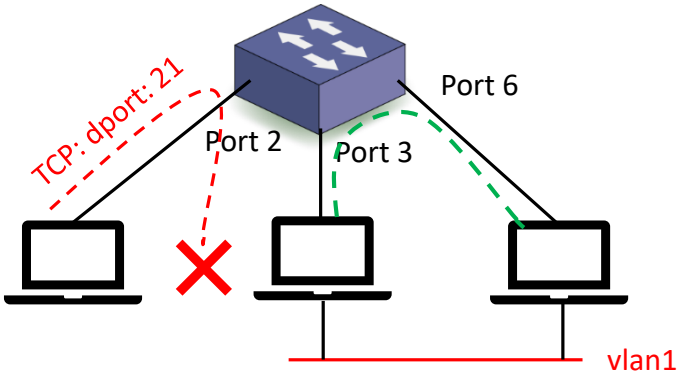
SFC Issues

- **Ordering and Application of SFs**
- **Security Considerations**
- Topological Dependencies
- Configuration Complexity
- Constrained SF availability
- Transport Dependencies
- Traffic Selection Criteria

CONTRIBUTION

- SUPC utilizes packet header fields and traffic steering of SFs and to composes a set of OpenFlow rules with no duplicates.
- SFC Ordering and Placement Preservation using SFC Composition.
- SUPC identifies four major type of rule conflicts based on important network and security properties.

OpenFlow Table and Flow Rule



Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

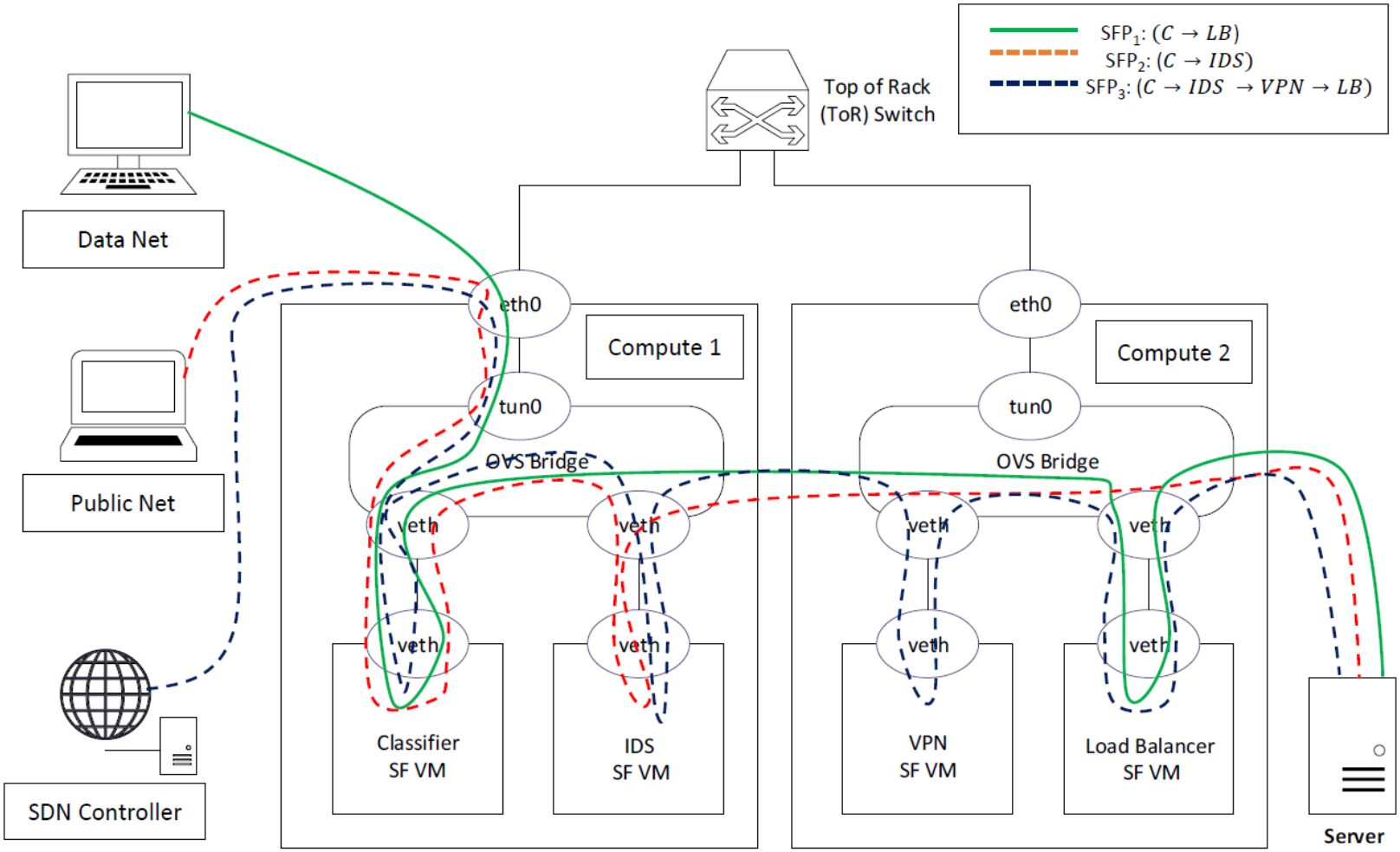
Flow Rule: $r_i = (p_i, \rho_i, h_i, a_i, s_i)$

Field	Interpretation
p_i	Rule Priority
ρ_i	Traffic Protocol – TCP/UDP
$h_i = (\alpha_{s_i}, \alpha_{d_i}, \beta_{s_i}, \beta_{d_i}, \gamma_{s_i}, \gamma_{d_i})$	Packet header = (L2 srcmac, L2 dstmac, L3 srcip, L3 dstip, L4 sport, L4 dport)
a_i	Action for the flow rule
s_i	Flow rule Statistics – packets/sec, bytes/sec.

SFC Requirements

1. Traffic coming into the network should be classified into different categories based on source IP address using Classifier SF.
2. Any traffic not part of data network security domain should be processed via Intrusion Detection System.
3. Data network traffic and SDN controller traffic should go through Load balancing SF.
4. Control plane traffic from SDN controller should be encrypted using public key encryption scheme.

SFC Example in Cloud



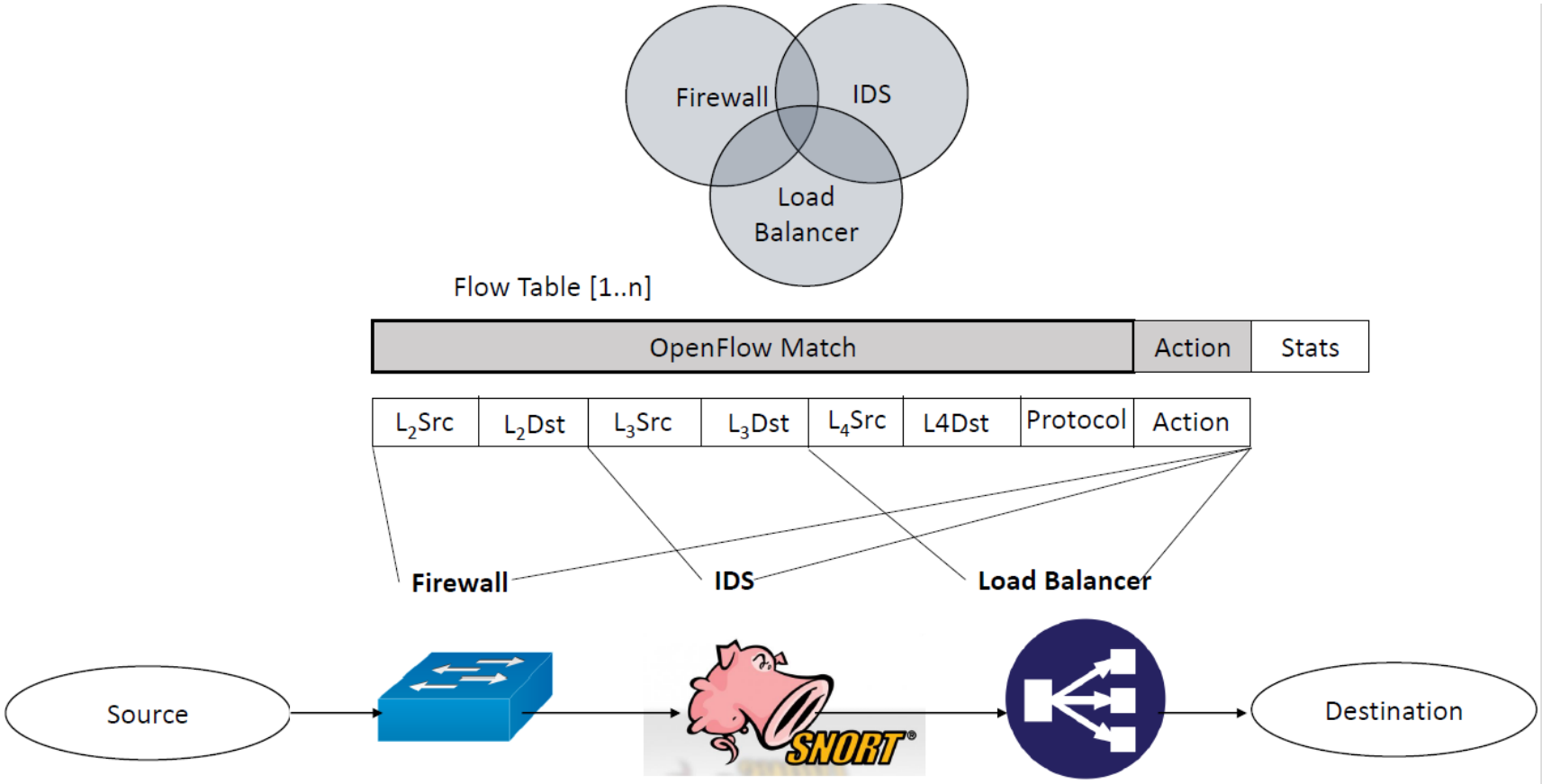
SFC Deployment Strategies

Strategy 1 Order: C → VPN → IDS → LB.

Strategy 2 Order: C → LB → IDS → VPN.

Strategy 3 Order: C → IDS → VPN → LB.

SF Placement Issue



SFC Flow Composition Analysis

action	protocol	srcip	srcport	dstip	dstport	
LOG	*tcp*	*192.168.1.0/24*	*any*	→ *any*	*!6000-6010*	*- (1) IDS Rule Format*
						- (2) IDS Rule
target	**protocol**	**opt**	**in**	**out**	**srcip**	**dstip**
DROP	*tcp*	*--*	***	***	*192.168.1.0/28*	***

- (3) Firewall Rule Format

-(4) Firewall Rule

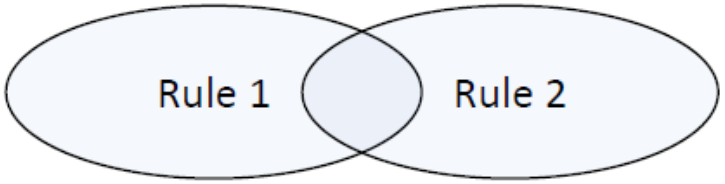
Source	*Protocol*	*L_2Src*	*L_2Dst*	*L_3Src*	*L_3Dst*	*L_4Src*	*L_4Dst*	*Action*	*Priority*
IDS	tcp	*	*	192.168.1.0/24	*	*	!6000-6010	LOG	1
Firewall	tcp	*	*	192.168.1.0/28	*	*	*	DROP	2

(a) OpenFlow rules before priority inversion

Source	*Protocol*	*L_2Src*	*L_2Dst*	*L_3Src*	*L_3Dst*	*L_4Src*	*L_4Dst*	*Action*	*Priority*
Firewall	tcp	*	*	192.168.1.0/24	*	*	*	DROP	1
IDS	tcp	*	*	192.168.1.0/24	*	*	!6000-6010	LOG	2

(b) OpenFlow rules after priority inversion

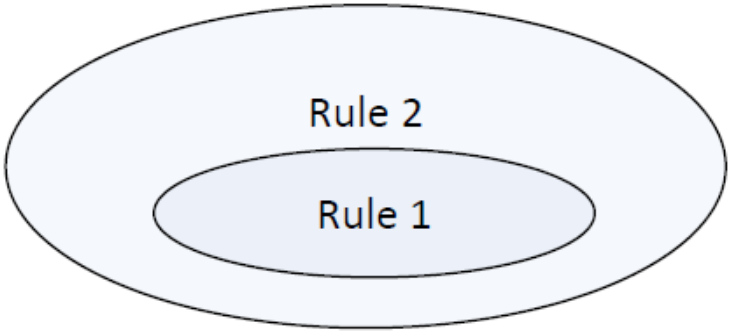
SFC Flow Conflict Analysis



(a) Intersection

$$\{match(R_1) \cap match(R_2)\} \,\widehat{\&}\, \{A(R_1) \neq A(R_2)\}$$

$$\{match(R_1) \cap match(R_2)\} \,\widehat{\&}\, \{A(R_1) = A(R_2)\}$$

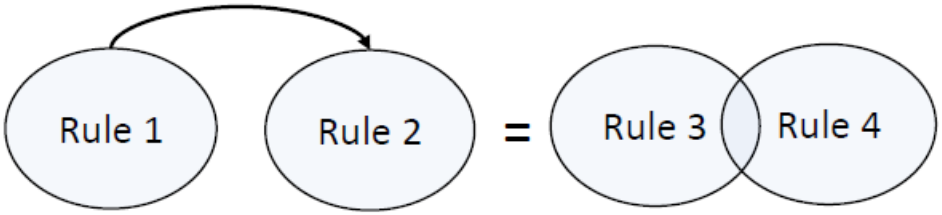


(b) Subsumption

$$\{match(R_1) \subseteq match(R_2)\} \,\widehat{\&}\, \{A(R_1) \neq A(R_2)\}$$

$$\{match(R_1) \subseteq match(R_2)\} \,\&\, \{A(R_1) = A(R_2)\}$$

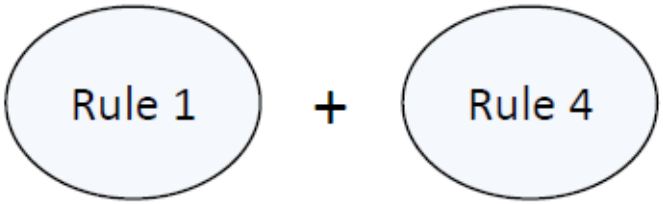
SFC Flow Conflict Analysis



(c) Transitivity

$$\{match(R_1), A(R_1) \rightarrow match(R_2), A(R_2) = match\ (R_3), A(R_3)\}$$

$$\{match(R_3) \cap match(R_4)\}\ \widehat{\&}\ \{A(R_3) \neq A(R_4)\}$$



(d) Symmetry

$$P(Symmetry) \leftarrow \{match(R_1), A(R_1)\} \cup \{match(R_2), A(R_2)\}$$

Ira A. Fulton Schools of Engineering
ARIZONA STATE UNIVERSITY

Flow Rule Conflict Example

Rule-ID	Prot	L2 Src	L2 Dst	L3 Src	L3 Dst	L4 Src	L4 Dst	Action
1	tcp	*	*	192.168.1.0/24	192.168.2.20	*	*	ALLOW
2	tcp	*	*	192.168.1.16	192.168.2.0/24	*	*	ALLOW
3	tcp	*	*	192.168.1.18	192.168.2.0/24	*	*	DENY
4	tcp	*	*	192.168.1.0/24	192.168.2.0/28	*	*	ALLOW
5	tcp	*	*	192.168.1.0/28	192.168.2.0/28	443	443	DENY
6	tcp	*	*	192.168.2.0/24	192.168.3.0/24	*	*	ALLOW
7	tcp	*	*	192.168.1.0/24	192.168.3.0/24	80	80	DENY
8	tcp	*	*	192.168.2.0/24	192.168.1.0/24	*	*	ALLOW
9	tcp	*	*	192.168.2.12	192.168.1.0/24	*	80	DENY

EXPERIMENTAL ANALYSIS

Rule Composition Analysis

Time (s)	IDS+Netfilter Rules	Flow Rules
5	2056	54
10	4014	85
15	7166	104
20	9686	171
25	12241	179
30	13472	201

Table I

IDS AND NETFILTER OPENFLOW RULE COMPOSITION

SCALABILITY OF RULE COMPOSITION ALGORITHM

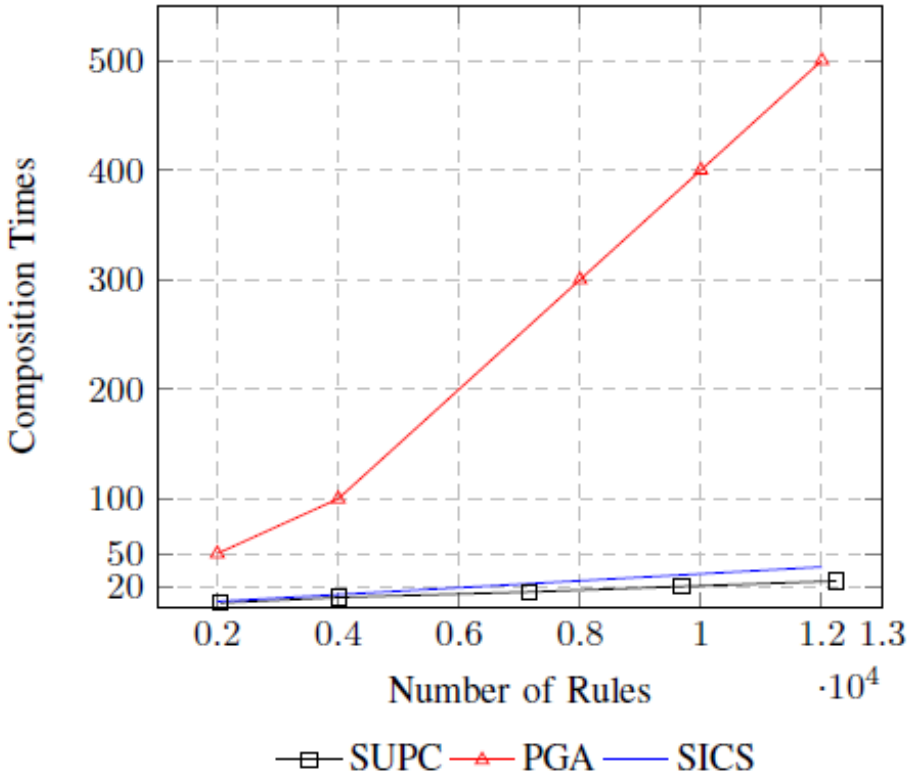


Figure 6. Number of Rules vs Composition Time - SUPC, PGA [8], SICS [11]

FLOW RULE CONFLICT ANALYSIS RESULTS

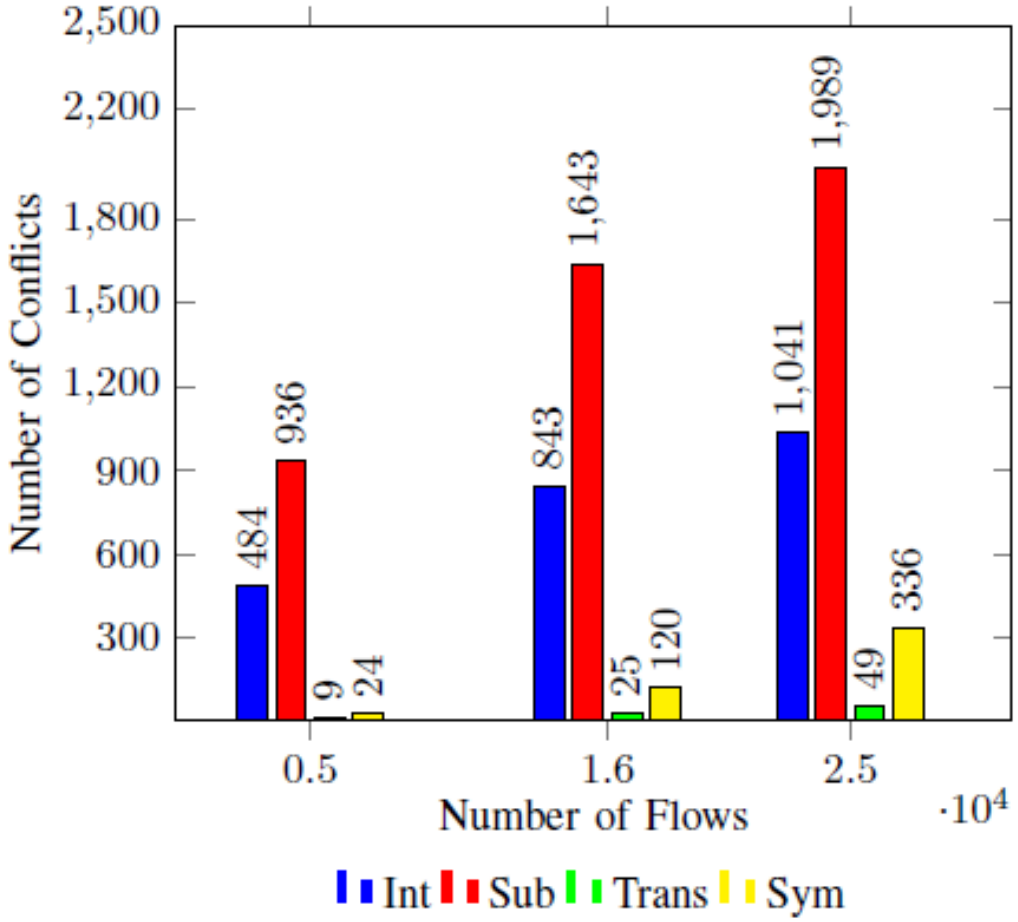


Figure 7. Number of Conflicts in SFC

CONCLUSION

- SUPC translates traffic and security policies of various SF into common OpenFlow format.
- Our experimental results on the dataset of Netfilter firewall rules and Bro IDS achieved a significant reduction in matching rules.
- SUPC identified four class of conflicts among the rules of various SFs which can cause security violations and service disruption.

References

- Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." *Proceedings of the IEEE* 103.1 (2015): 14-76
- Fayazbakhsh, Seyed Kaveh, et al. "Flowtags: Enforcing network-wide policies in the presence of dynamic middlebox actions." *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013..
- Gember-Jacobson, Aaron, et al. "OpenNF: Enabling innovation in network function control." *ACM SIGCOMM Computer Communication Review*. Vol. 44. No. 4. ACM, 2014.
- Gember-Jacobson, Aaron, et al. "OpenNF: Enabling innovation in network function control." *ACM SIGCOMM Computer Communication Review*. Vol. 44. No. 4. ACM, 2014.
- Joseph, Dilip A., Arsalan Tavakoli, and Ion Stoica. "A policy-aware switching layer for data centers." *ACM SIGCOMM Computer Communication Review*. Vol. 38. No. 4. ACM, 2008.
- McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." *ACM SIGCOMM Computer Communication Review* 38.2 (2008): 69-74.

References

- Pisharody, Sandeep, et al. "Brew: A security policy analysis framework for distributed SDN-based cloud environments." *IEEE Transactions on Dependable and Secure Computing* (2017).
- Prakash, Chaithan, et al. "Pga: Using graphs to express and automatically reconcile network policies." *ACM SIGCOMM Computer Communication Review*. Vol. 45. No. 4. ACM, 2015.
- Sendi, Alireza Shameli, et al. "Efficient provisioning of security service function chaining using network security defense patterns." *IEEE Transactions on Services Computing* (2016).
- Trajkovska, Irena, et al. "SDN-based service function chaining mechanism and service prototype implementation in NFV scenario." *Computer Standards & Interfaces* 54 (2017): 247-265.
- Wang, Huazhe, et al. "SICS: Secure In-Cloud Service Function Chaining." *arXiv preprint arXiv:1606.07079* (2016).

THANK YOU