Challenges and Preparedness of SDN-based Firewalls

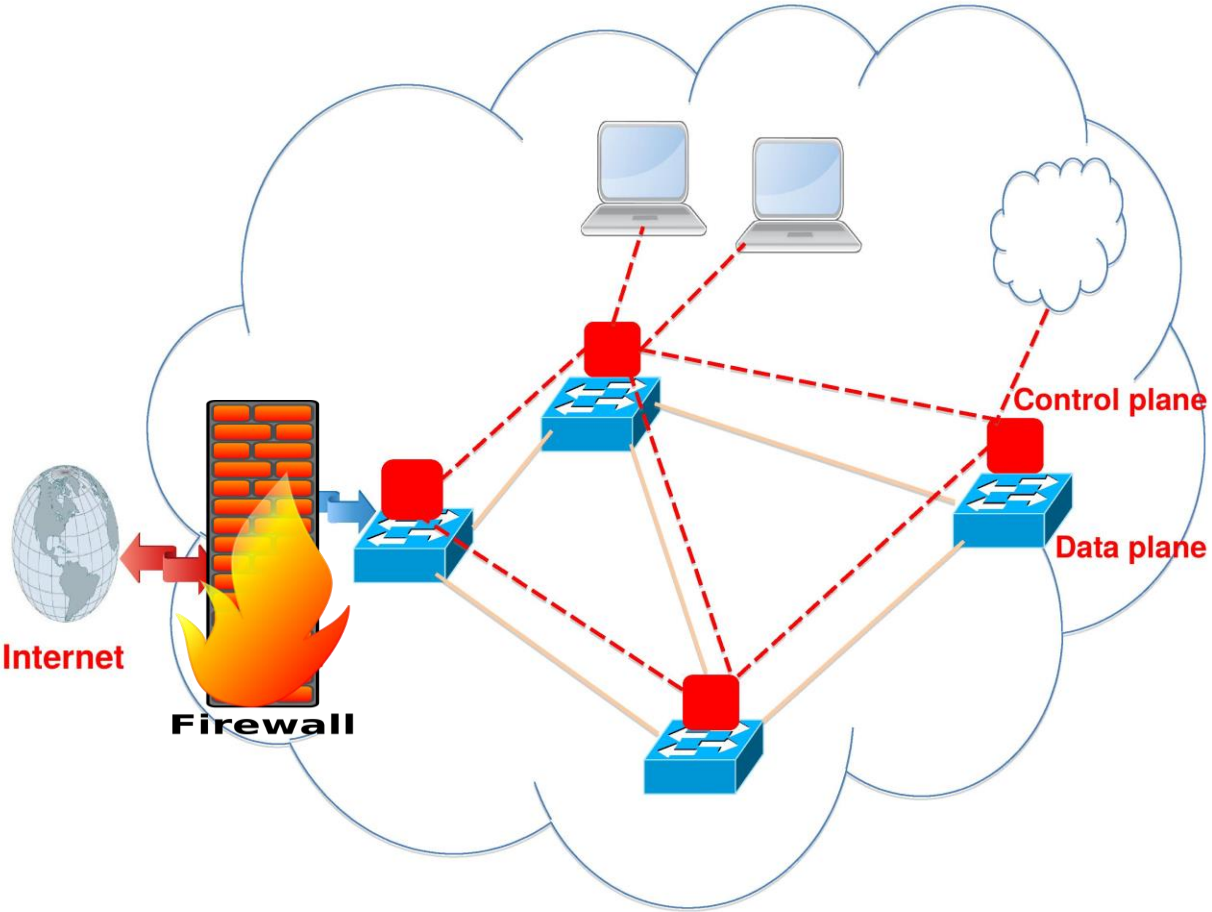
Vaibhav Hemant Dixit, Sukwha Kyung, Ziming Zhao, Adam Doupé, Yan Shoshitaishvili and Gail-Joon Ahn

ARIZONA STATE UNIVERSITY

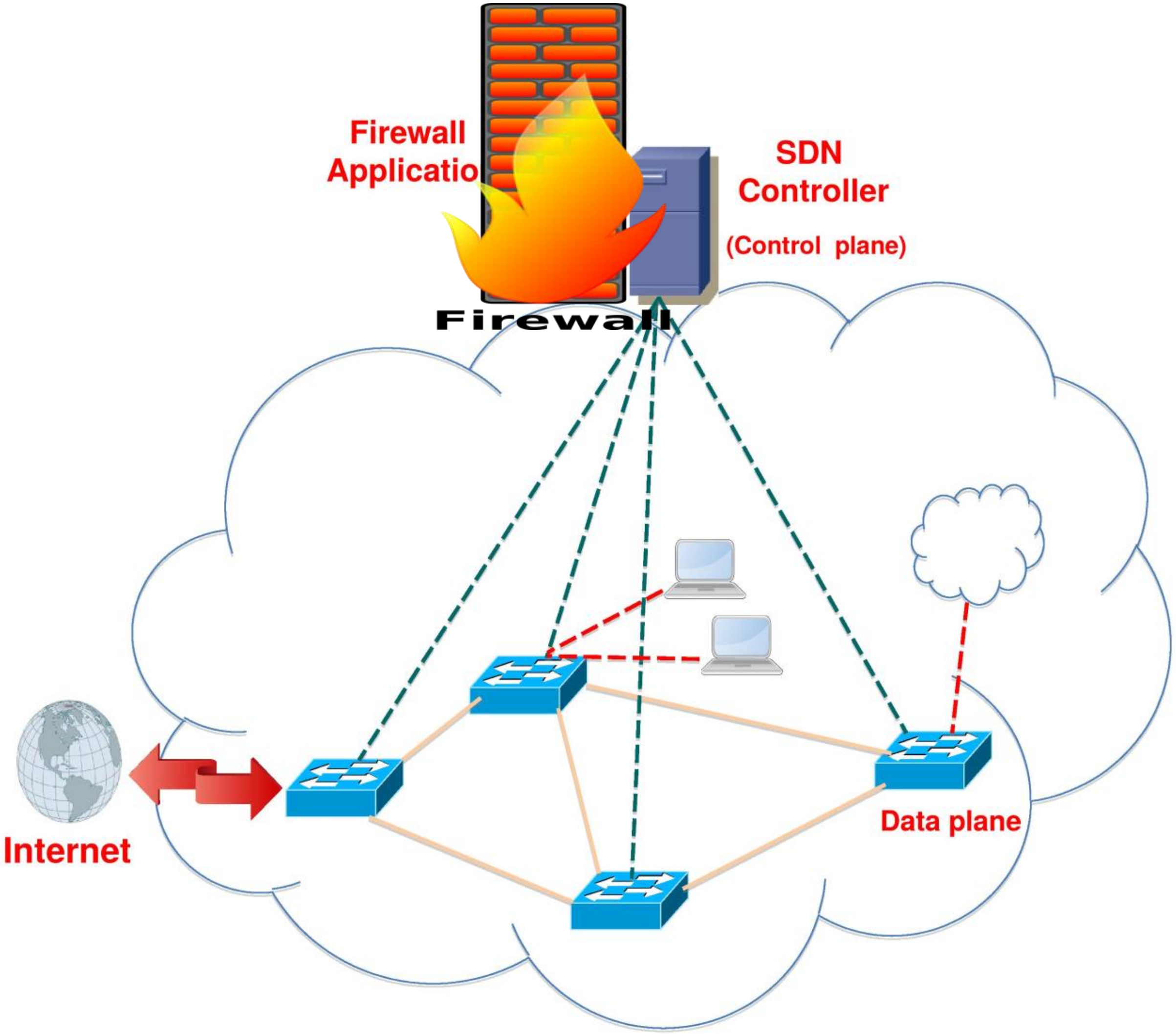
CENTER FOR CYBERSECURITY & DIGITAL FORENSICS

sefcom security engineering for future computing

Firewall setting in traditional network

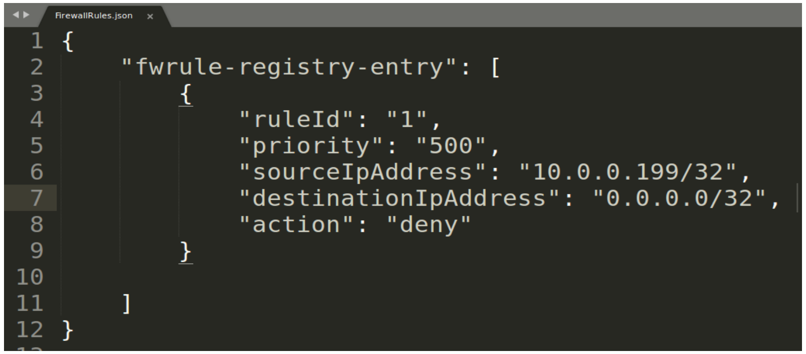


Firewall in SDN-based Network



SDN-based Firewall – Key concepts

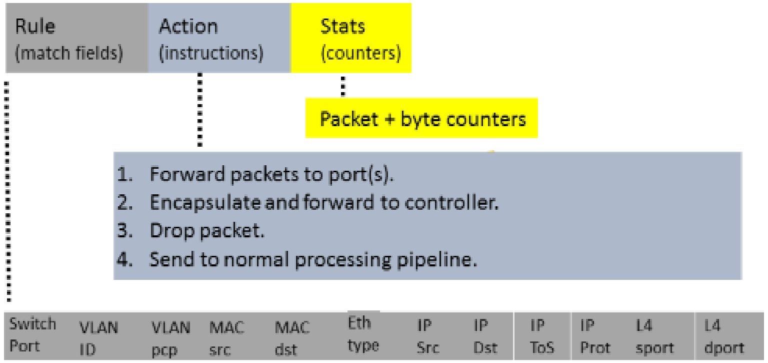
- Firewall Policy



```
{
    "fwrule-registry-entry": [
        {
            "ruleId": "1",
            "priority": "500",
            "sourceIpAddress": "10.0.0.199/32",
            "destinationIpAddress": "0.0.0.0/32",
            "action": "deny"
        }

    ]
}
```

- Flow Policy



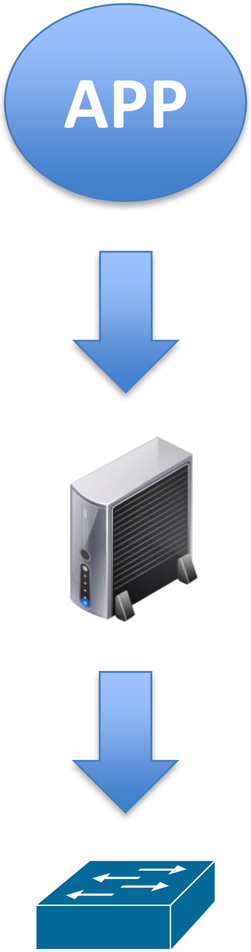
Rule (match fields) | Action (instructions) | Stats (counters)

Packet + byte counters

1. Forward packets to port(s).
2. Encapsulate and forward to controller.
3. Drop packet.
4. Send to normal processing pipeline.

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP ToS	IP Prot	L4 sport	L4 dport

- Conversion of policy to flow rules in switches

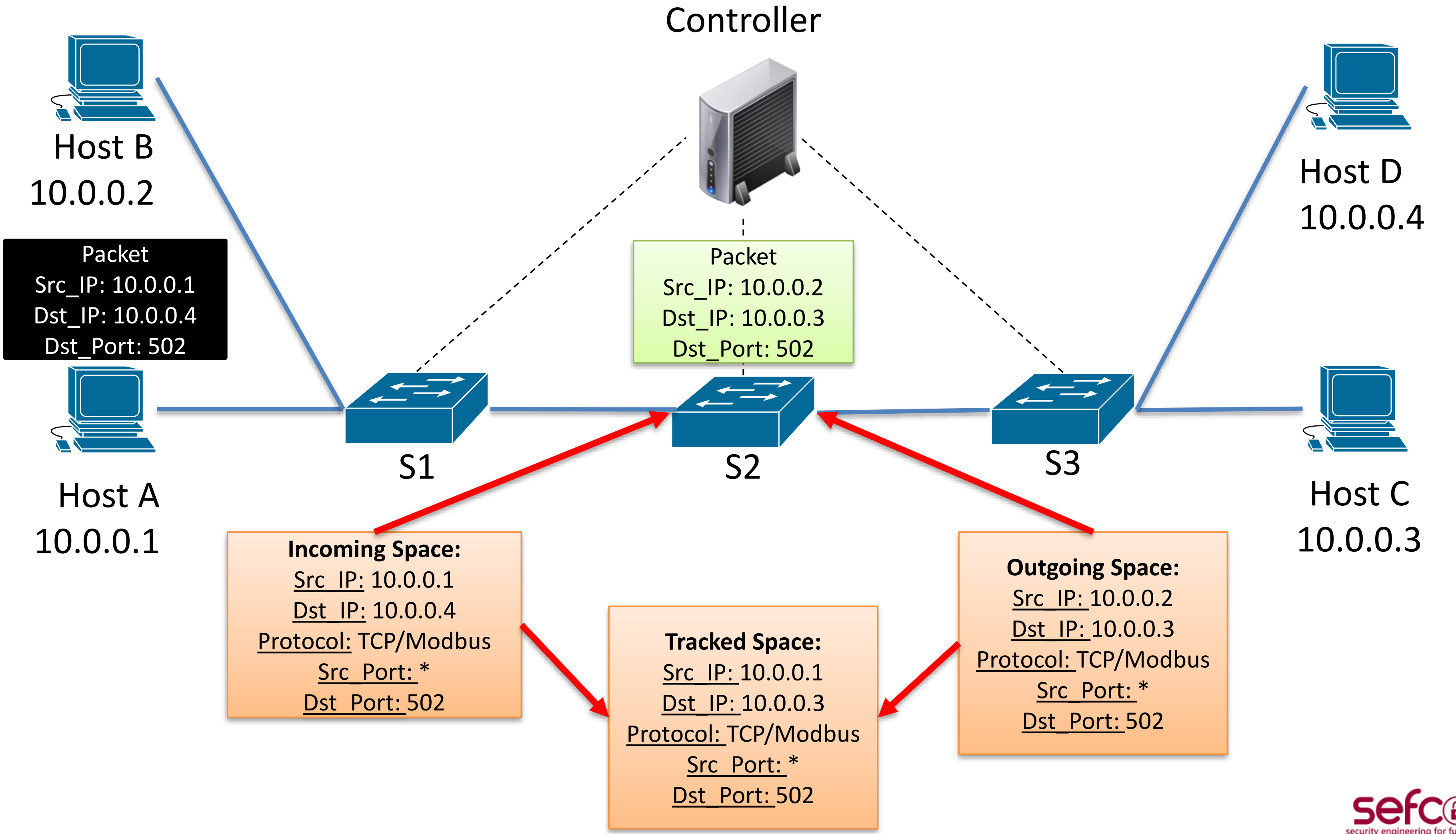


Promises of SDN-based firewalls

- Centralized Policy Enforcement
- Centralized Flow Tracking
- Centralized Conflict Resolution
- Scalability and Concurrency
- **Automatic Priority Handling**
- **Multi Tenant Support**
- **Stateful Support**

Promise:
Centralized Conflict Detection

Centralized Flow Tracking



Controller

Host B
10.0.0.2

Packet
Src_IP: 10.0.0.1
Dst_IP: 10.0.0.4
Dst_Port: 502

Host A
10.0.0.1

Packet
Src_IP: 10.0.0.2
Dst_IP: 10.0.0.3
Dst_Port: 502

Host D
10.0.0.4

Host C
10.0.0.3

S1

S2

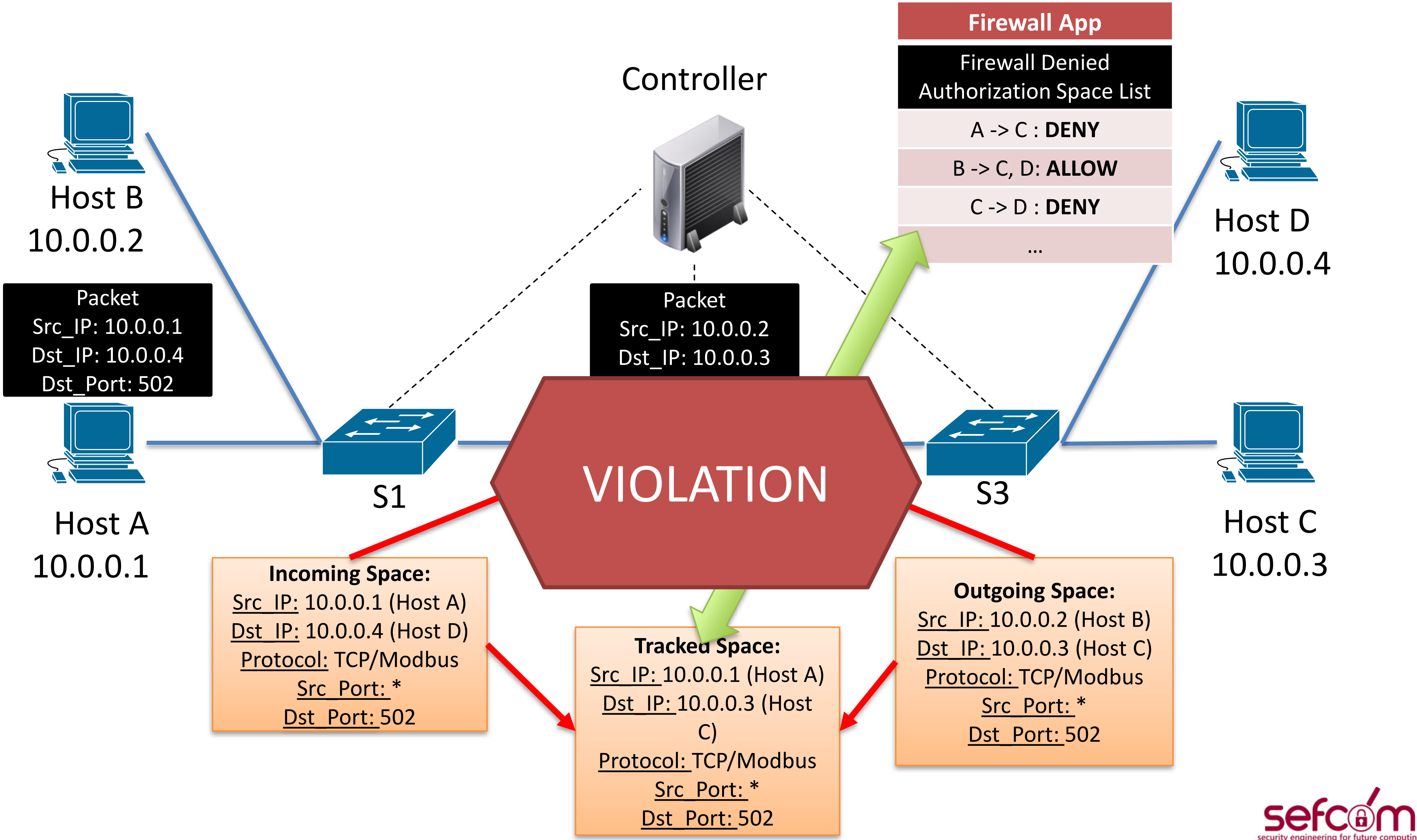
S3

Incoming Space:
Src_IP: 10.0.0.1
Dst_IP: 10.0.0.4
Protocol: TCP/Modbus
Src_Port: *
Dst_Port: 502

Tracked Space:
Src_IP: 10.0.0.1
Dst_IP: 10.0.0.3
Protocol: TCP/Modbus
Src_Port: *
Dst_Port: 502

Outgoing Space:
Src_IP: 10.0.0.2
Dst_IP: 10.0.0.3
Protocol: TCP/Modbus
Src_Port: *
Dst_Port: 502

Centralized Conflict Detection



Firewall App
Firewall Denied Authorization Space List
A -> C : **DENY**
B -> C, D: **ALLOW**
C -> D : **DENY**
…

Packet
Src_IP: 10.0.0.1
Dst_IP: 10.0.0.4
Dst_Port: 502

Packet
Src_IP: 10.0.0.2
Dst_IP: 10.0.0.3

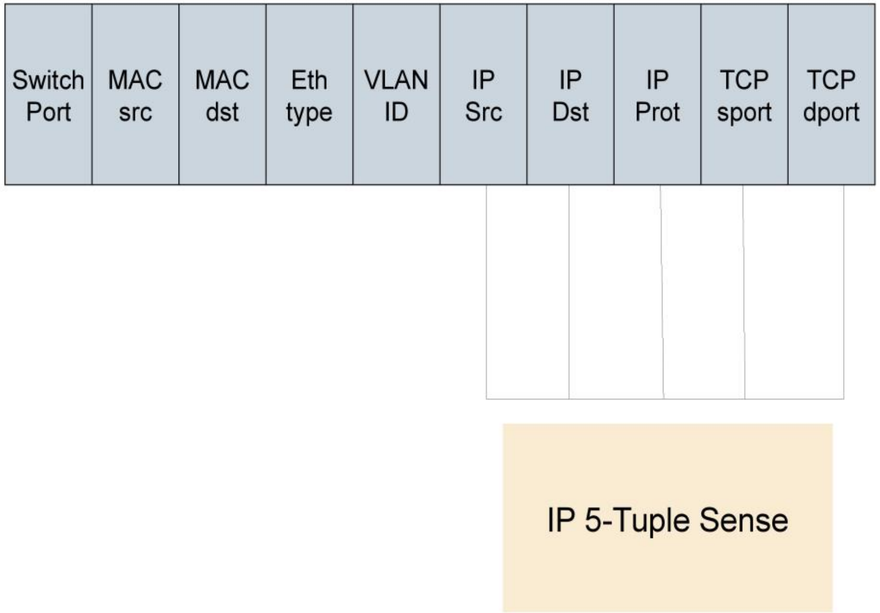
Incoming Space:
Src_IP: 10.0.0.1 (Host A)
Dst_IP: 10.0.0.4 (Host D)
Protocol: TCP/Modbus
Src_Port: *
Dst_Port: 502

Tracked Space:
Src_IP: 10.0.0.1 (Host A)
Dst_IP: 10.0.0.3 (Host C)
Protocol: TCP/Modbus
Src_Port: *
Dst_Port: 502

Outgoing Space:
Src_IP: 10.0.0.2 (Host B)
Dst_IP: 10.0.0.3 (Host C)
Protocol: TCP/Modbus
Src_Port: *
Dst_Port: 502

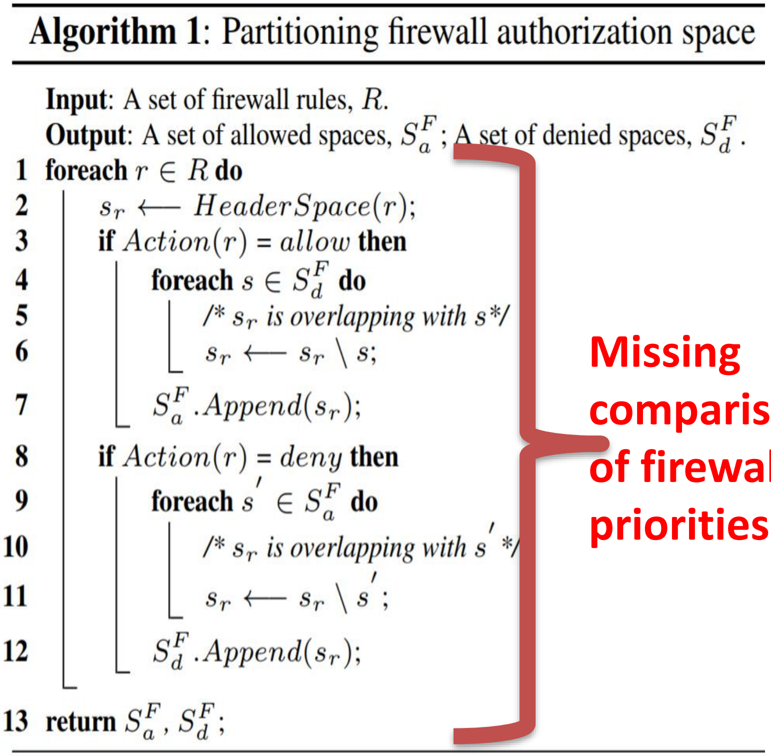
Challenges in Conflict Detection

- Ambiguous Flow Path Space calculation
- Disregard for order of rules and their priority



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

IP 5-Tuple Sense



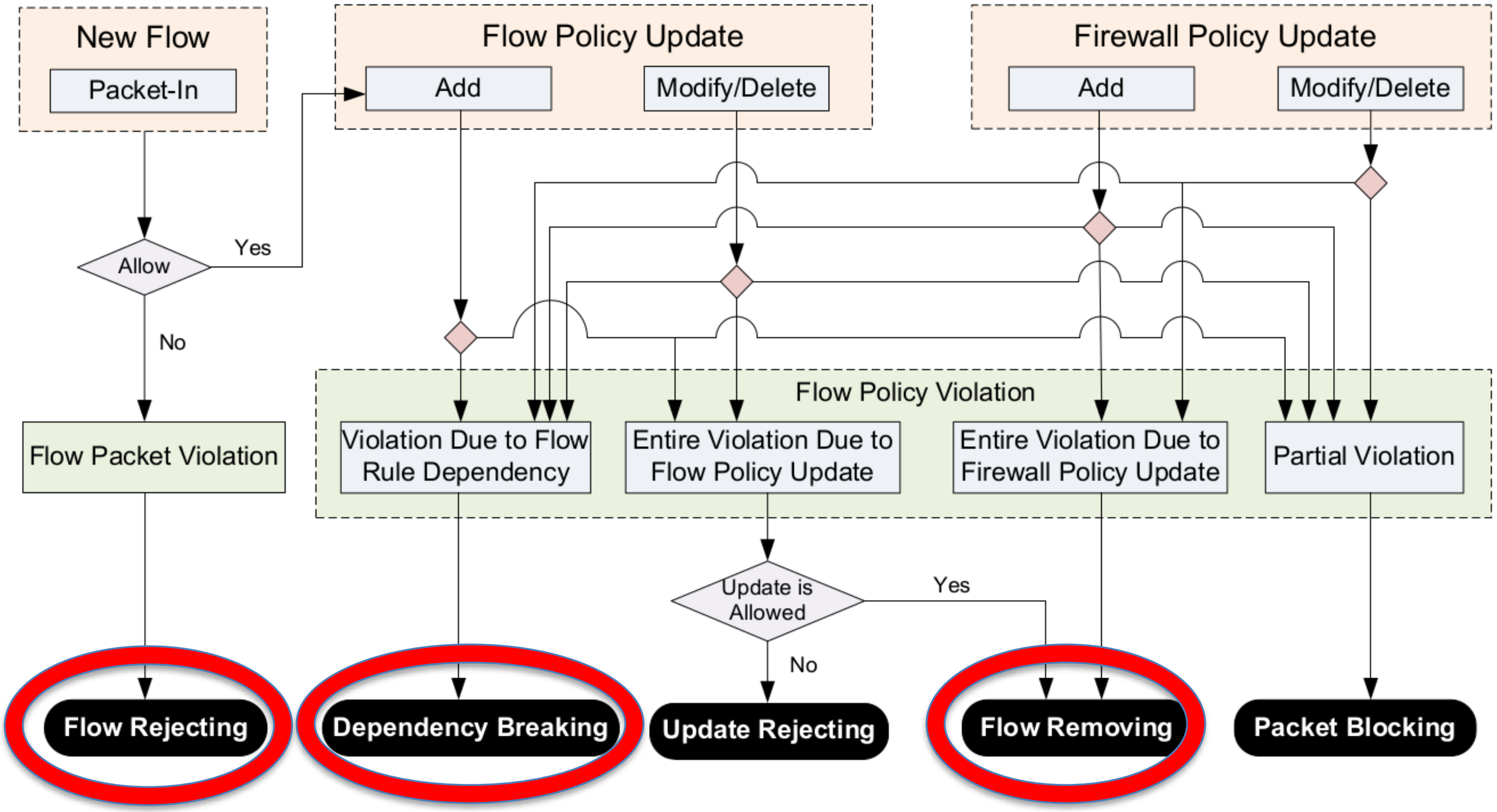
Algorithm 1: Partitioning firewall authorization space

```
Input: A set of firewall rules, $R$.
Output: A set of allowed spaces, $S_a^F$; A set of denied spaces, $S_d^F$.
foreach $r \in R$ do
    $s_r \longleftarrow HeaderSpace(r)$;
    if $Action(r) = allow$ then
        foreach $s \in S_d^F$ do
            /* $s_r$ is overlapping with $s$ */
            $s_r \longleftarrow s_r \setminus s$;
        $S_a^F.Append(s_r)$;
    if $Action(r) = deny$ then
        foreach $s' \in S_a^F$ do
            /* $s_r$ is overlapping with $s'$ */
            $s_r \longleftarrow s_r \setminus s'$;
        $S_d^F.Append(s_r)$;
return $S_a^F, S_d^F$;
```

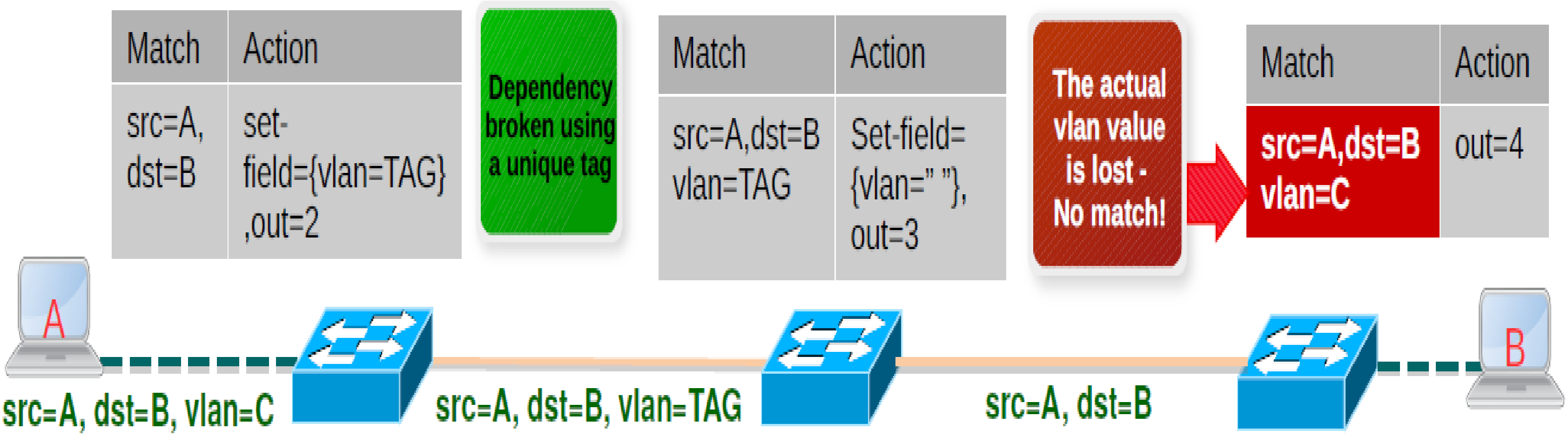
Missing comparison of firewall priorities

Promise:
Centralized Conflict Resolution

Centralized Conflict Resolution



Mistakes in Conflict Resolution - 1



Match	Action
src=A, dst=B	set-field={vlan=TAG} ,out=2

Dependency broken using a unique tag

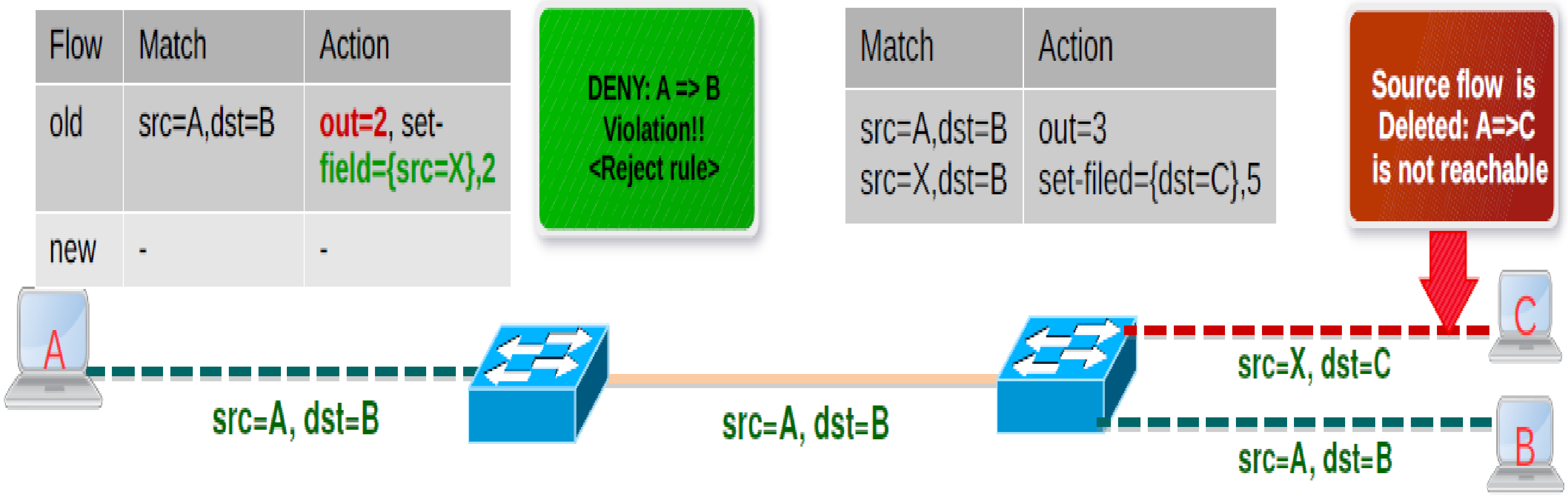
Match	Action
src=A,dst=B vlan=TAG	Set-field= {vlan=" "}, out=3

The actual vlan value is lost - No match!

Match	Action
src=A,dst=B vlan=C	out=4

(a) Resolution method: Dependency Breaking using Flow-Tagging

Mistakes in Conflict Resolution - 2



Flow	Match	Action
old	src=A,dst=B	out=2, set-field={src=X},2
new	-	-

DENY: A => B
Violation!!
<Reject rule>

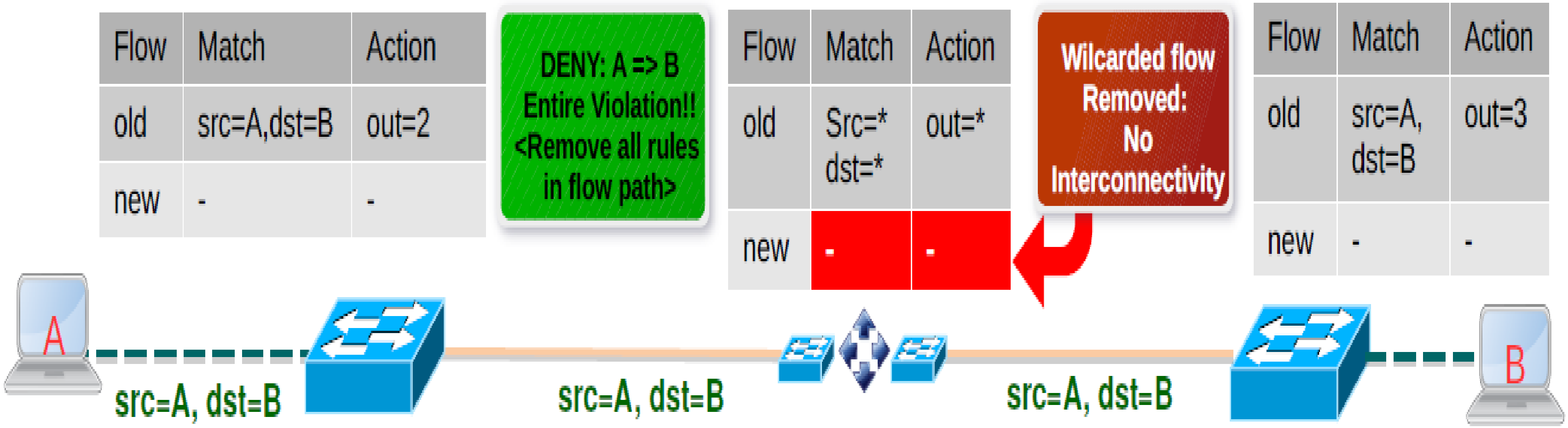
Match	Action

src=A,dst=B
src=X,dst=B
	out=3
	set-filed={dst=C},5

Source flow is Deleted: A=>C is not reachable

(b) Resolution method: Flow Rejection

Mistakes in Conflict Resolution - 3



Flow	Match	Action
old	src=A,dst=B	out=2
new	-	-

DENY: A => B
Entire Violation!!
<Remove all rules
in flow path>

Flow	Match	Action

old	Src=*
	dst=*
		out=*

new	-	-

Wilcarded flow
Removed:
No
Interconnectivity

Flow	Match	Action

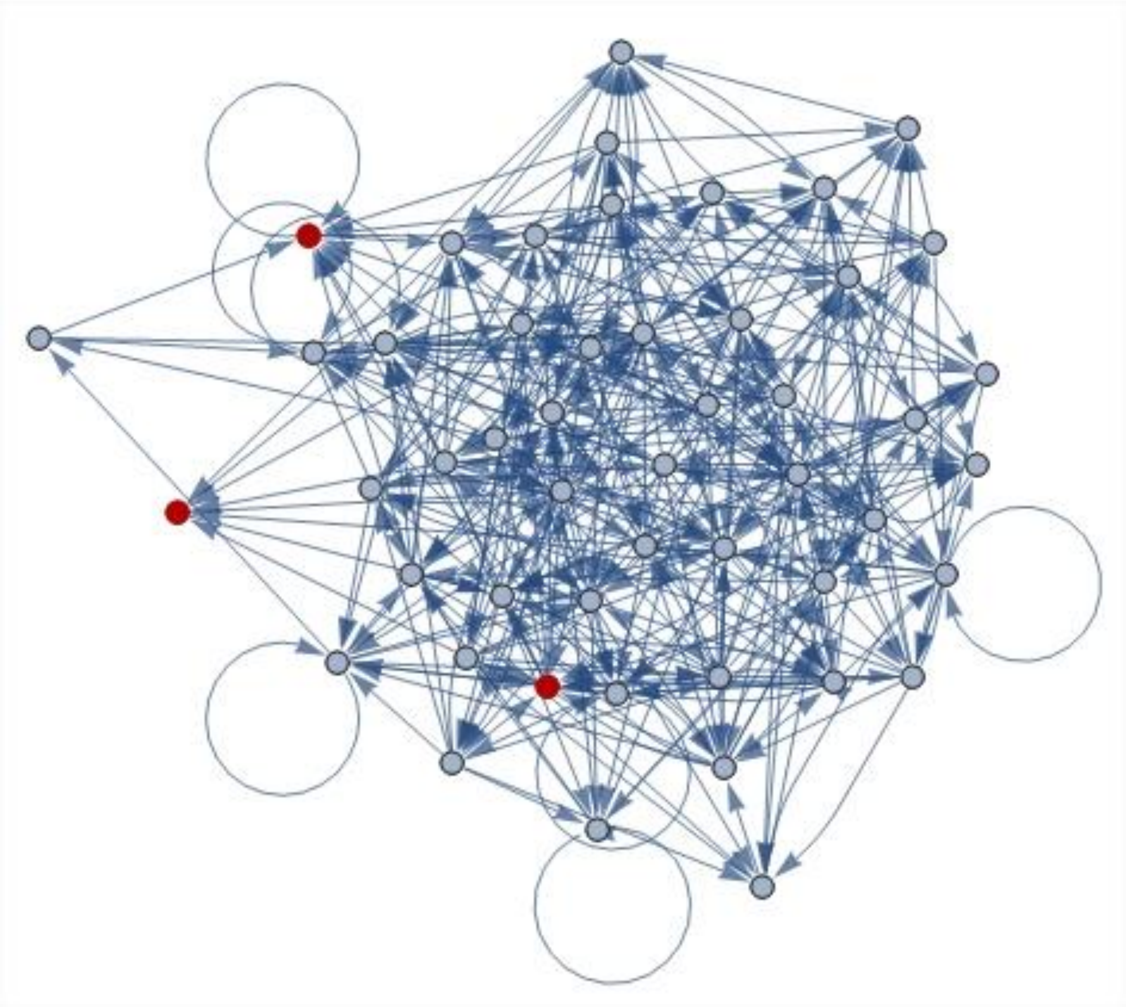
old	src=A,
	dst=B
		out=3

new	-	-

(c) Resolution method: Flow Removal

Promise:
Dynamic Network Scalability

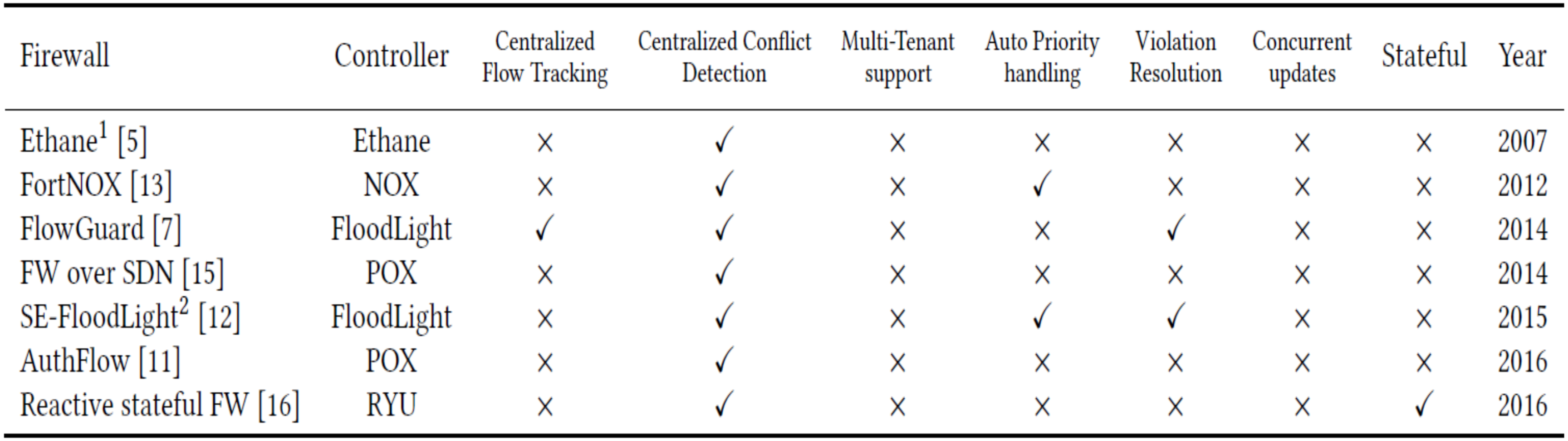
Scaling Networks



Challenges on Scaling Networks

- Multi-tenant networks
  - Use ten-tuple addressing.
- Significant increase in response time for a complicated plumbing graph
  - Detection and resolution in order of seconds!
  - Can be improved using ***reachability map***.
- Disregard for concurrent updates
  - Consider role-based access control.

Problems with current implementations



Firewall	Controller	Centralized Flow Tracking	Centralized Conflict Detection	Multi-Tenant support	Auto Priority handling	Violation Resolution	Concurrent updates	Stateful	Year
Ethane[1] [5]	Ethane	×	✓	×	×	×	×	×	2007
FortNOX [13]	NOX	×	✓	×	✓	×	×	×	2012
FlowGuard [7]	FloodLight	✓	✓	×	×	✓	×	×	2014
FW over SDN [15]	POX	×	✓	×	×	×	×	×	2014
SE-FloodLight[2] [12]	FloodLight	×	✓	×	✓	✓	×	×	2015
AuthFlow [11]	POX	×	✓	×	×	×	×	×	2016
Reactive stateful FW [16]	RYU	×	✓	×	×	×	×	✓	2016

Challenges hampering adoption of SDN-based firewalls.

Conclusion

- Defined the potential and capabilities that can be leveraged.
- Explored *challenges* faced by SDN-based firewalls approaches.
- Compared existing SDN-based firewall solutions against the key criteria.
- Proposed considerations for improvement.

Thank you !!

Vaibhav Hemant Dixit

vaibhav@asu.edu

sefcom.asu.edu

This work was partially supported by grant from National Science Foundation (NSF-ACI-1642031)