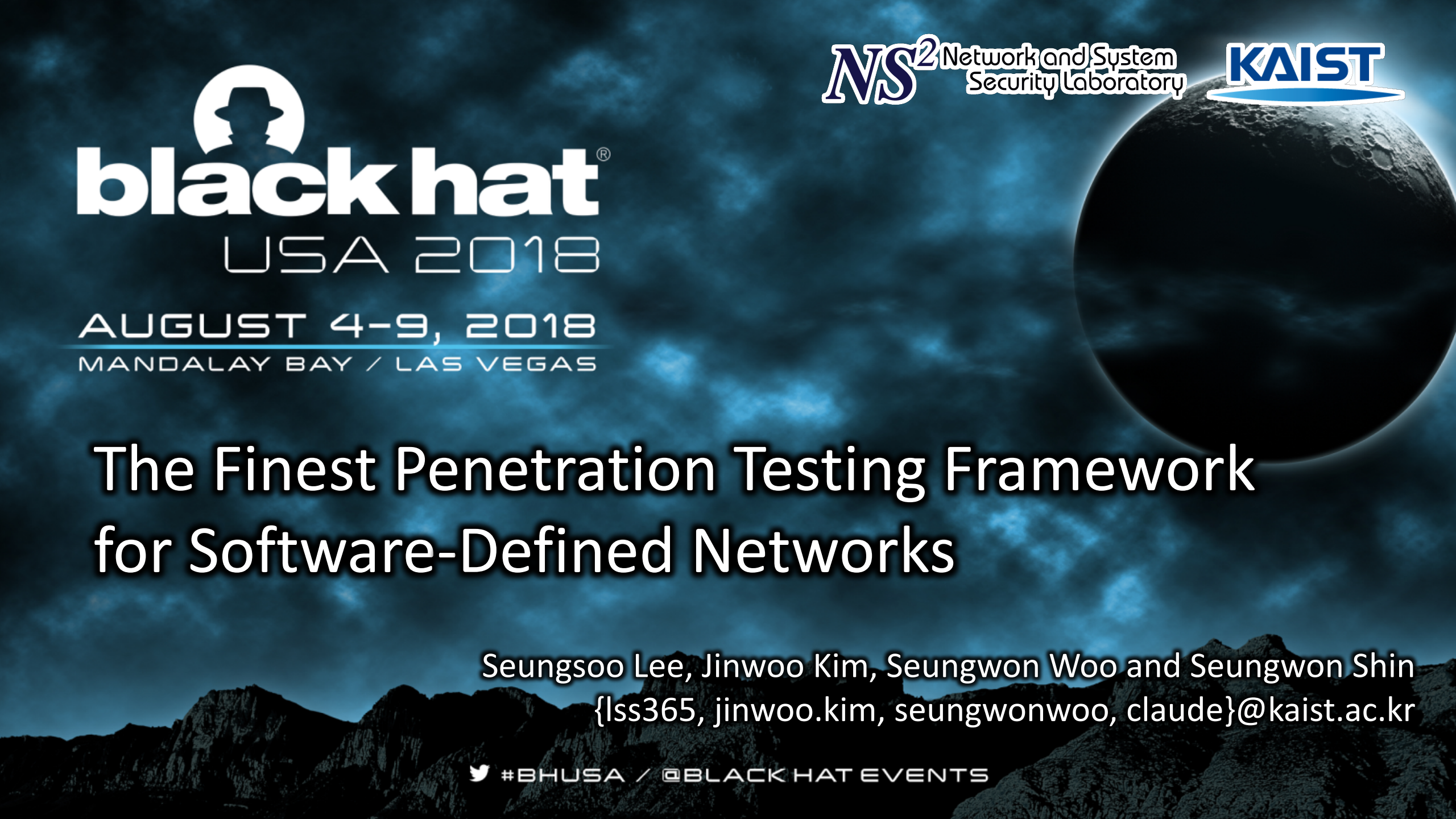


The Finest Penetration Testing Framework for Software-Defined Networks

Seungsoo Lee, Jinwoo Kim, Seungwon Woo and Seungwon Shin
{lss365, jinwoo.kim, seungwonwoo, claude}@kaist.ac.kr

#BHUSA / @BLACK HAT EVENTS

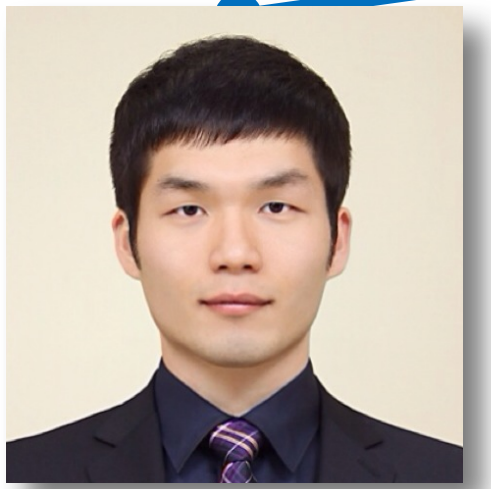
About us

Seungwon Shin

- Associate Professor of EE dept. at KAIST
- Leading Network and System Security Lab.

NS² Network and System Security Laboratory

KAIST



Seungsoo Lee

- PhD student at KAIST

Jinwoo Kim

- PhD student at KAIST

Seungwon Woo

- Master student at KAIST

Contents

1. Motivation of DELTA
2. Software-Defined Networking (SDN)
   - SDN & OpenFlow basics
   - Security of SDN
3. DELTA framework
   - Architecture
   - Attack case demonstrations
4. Final remarks

Motivation of DELTA

- Why needed?
  - **Software-defined Networking (SDN)** are still prone to security threats
  - We need to run security tests against our SDNs
  - But, manually testing each attack is time consuming and annoying job
- DELTA can **AUTOMATICALLY**…
  - Construct an SDN security test environment
  - ***(i) Reproduce the known attacks***
  - ***(ii) Find new attacks by randomizing SDN control flows (i.e., OpenFlow)***

Limitations of Traditional Networks

Image source: https://peterskastner.wordpress.com/category/complexity/

Expensive network/security devices (CAPEX)

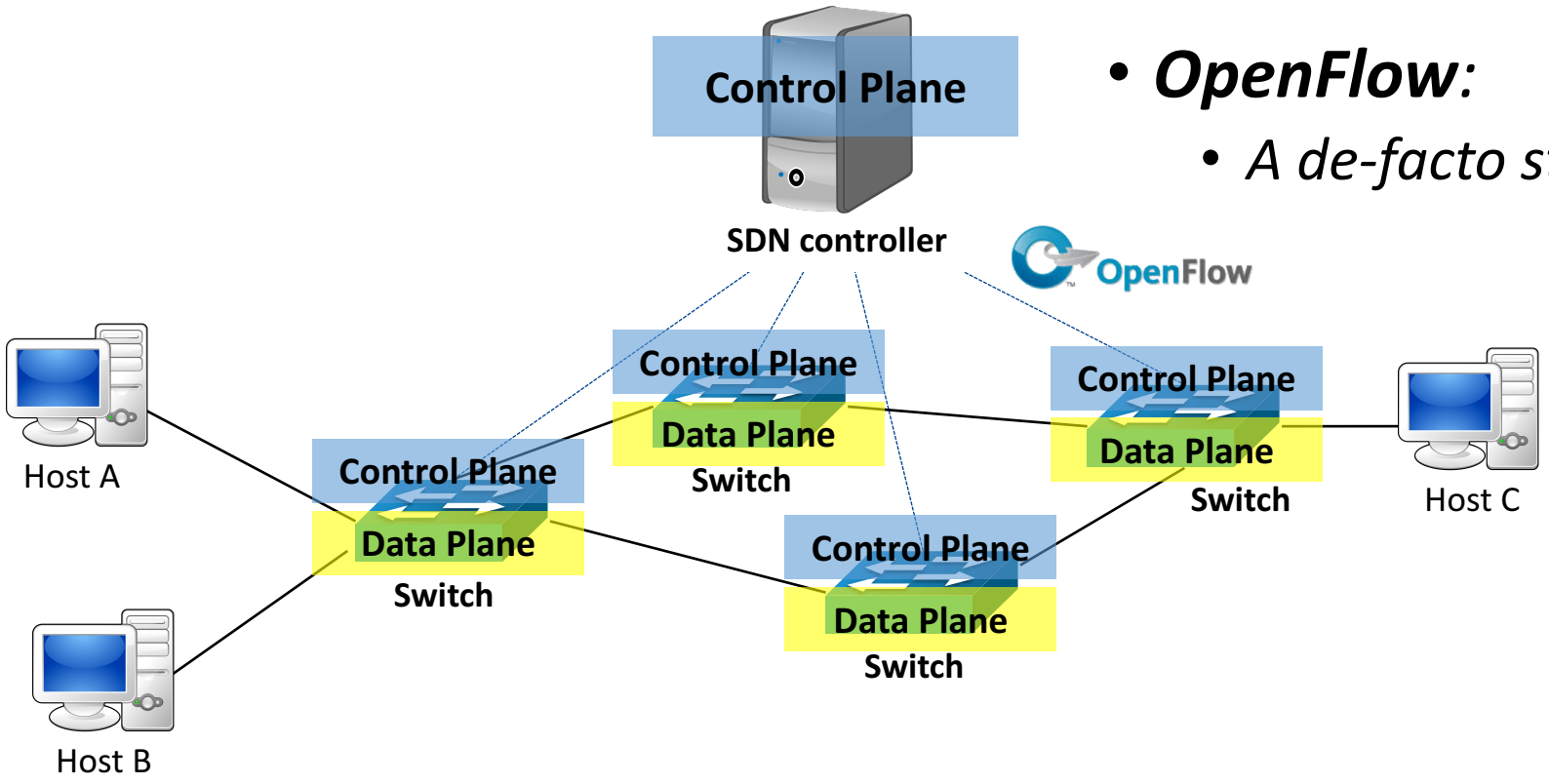
Proprietary firmware
Specialized hardware
…

Increased complexity of network management (OPEX)

Complicated maintenance
Manual configuration
…

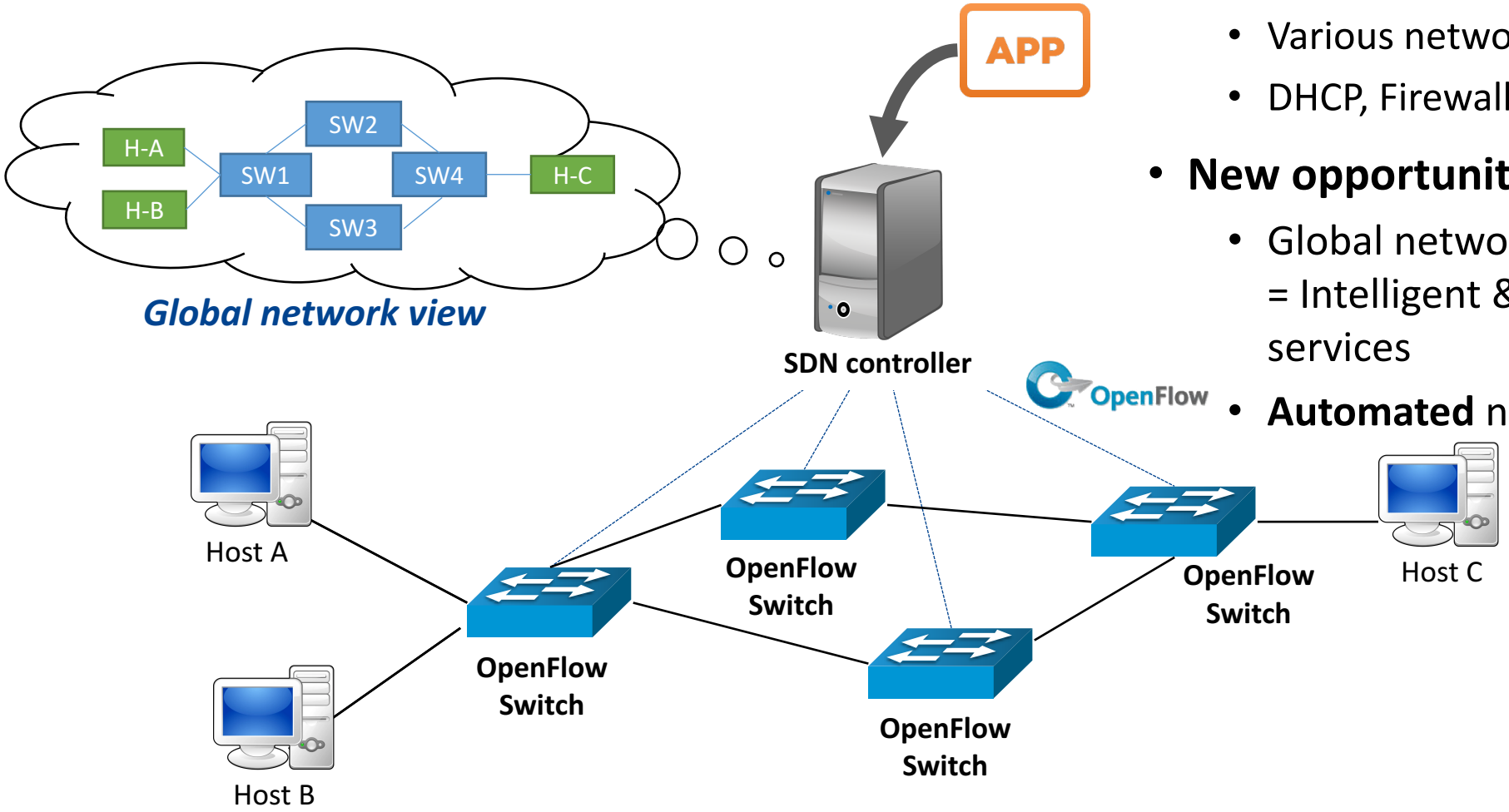
Software-defined Networking (SDN)

- ***Separation & Centralization*** of the control plane



- ***OpenFlow:***
  - *A de-facto standard SDN protocol*

SDN: New opportunities



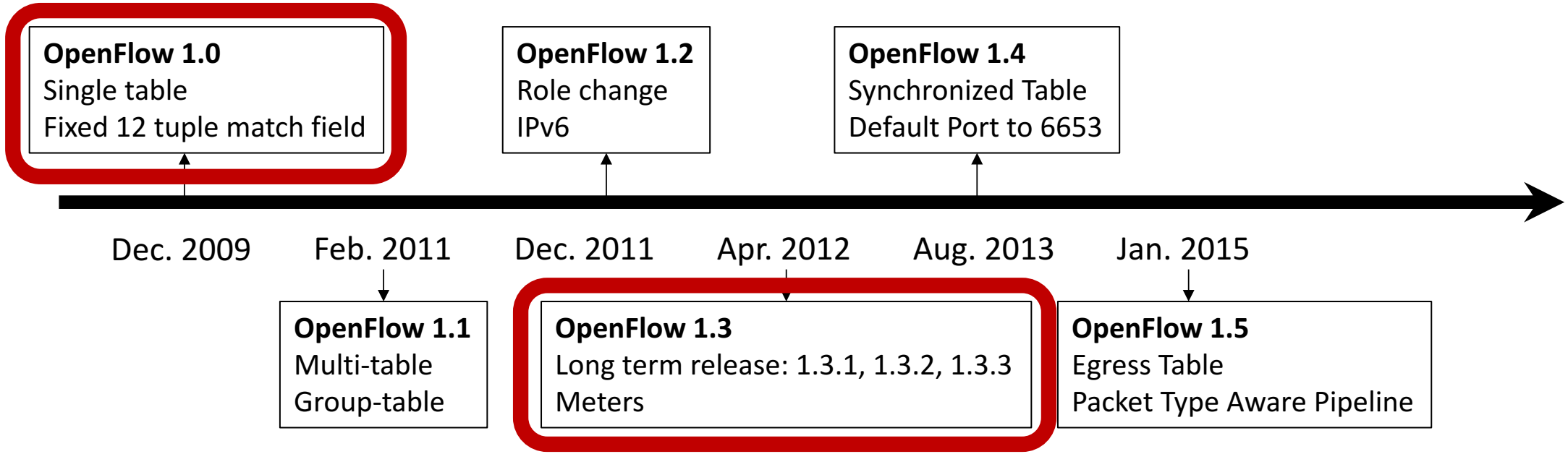
- **Flexible** service customization
  - Various network functions in SDN APPs
  - DHCP, Firewall, DDoS detector, and etc.
- **New opportunities**
  - Global network view + SDN APPs = Intelligent & innovative network/security services
- **Automated** network management

- *Global network view + APPs + Dynamic network control* = ***PROGRAMMABLE NETWORK !!***

OpenFlow

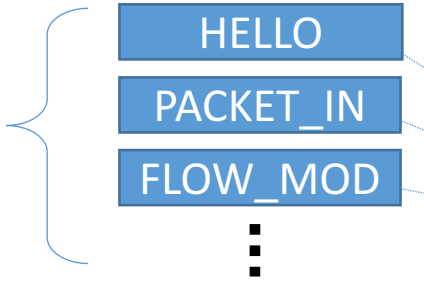
- A De-facto standard protocol in SDN
- Maintained by ***Open Networking Foundation***
- Supported by 120+ industrial members
- Version timeline

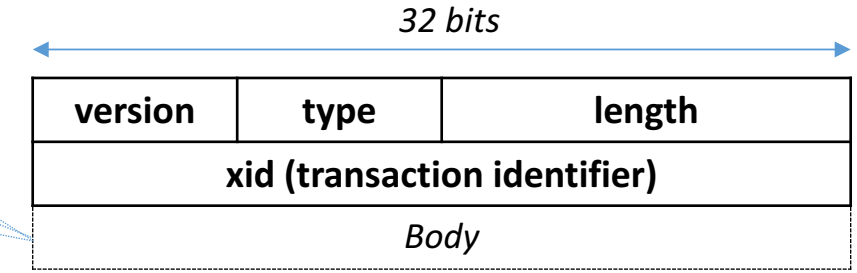
Open Networking Foundation
https://www.opennetworking.org/



OpenFlow 1.0

- **22** message types
- Flow table structure
  - Header fields, actions and counters





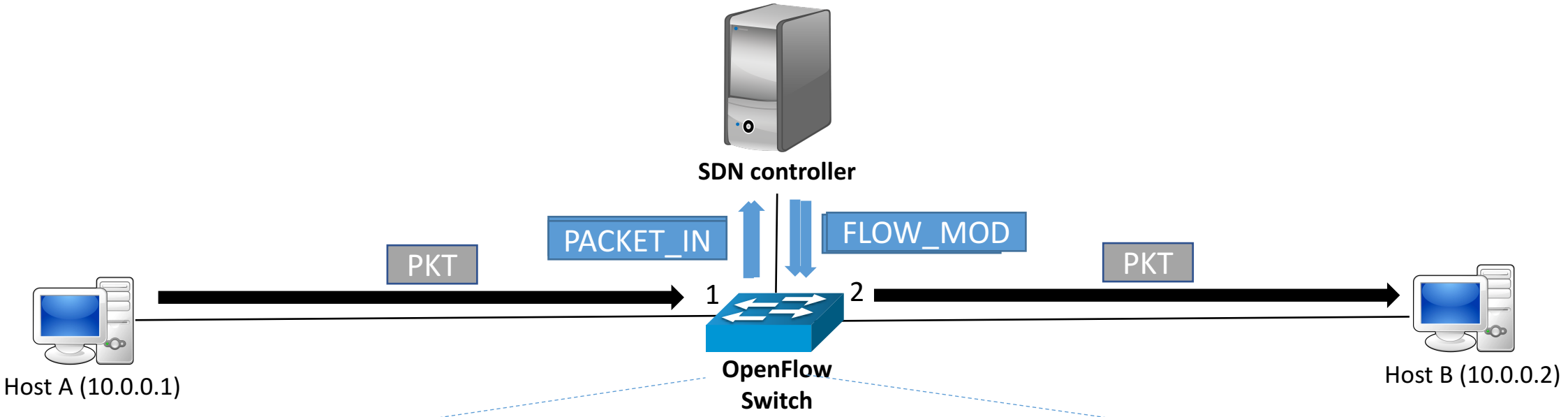
OpenFlow Structure

- Fixed 12 match fields
- If matched, perform a
- Forwa
- Drop
- Modify fields
- Per-table, per-flow, per-port and per queue
- Packet and byte counters

Header Fields (i.e., Match fields)												Actions	Counters
InPort	EthSrc	EthDst	EthType	VLANID	VLANPri	IPSrc	IPDst	IPProto	IPToS	TCP/UDP SrcPort	TCP/UDP DstPort		

Flow Table Structure

OpenFlow 1.0: Basic Operation



	Header Fields (i.e., Match fields)												Actions	Counters
Priority	InPort	EthSrc	EthDst	EthType	VLANID	VLANPri	IPSrc	IPDst	IPProto	IPToS	TCP/UDP SrcPort	TCP/UDP DstPort		
10	[**InPort**]: 1, [**EthType**]: 0x0800, [**IPDst**]: 10.0.0.2												Forward 2	***P: 1, B: 64***

Flow Table Structure

OpenFlow 1.0 vs. OpenFlow 1.3

OpenFlow 1.0

- Released in Dec. 2009.
- 22 message types
- Single controller
- Single flow table
- Fixed 12 tuple match fields

OpenFlow 1.3

- Released in Apr. 2012.
- **30** message types
- **Multiple** controllers
- **Multiple** flow tables
- Extensible match (OXM)
- **Group** table
- **Meter** table
- Instruction (action set)

SDN adoption: Enterprise

Deutsche Telekom, AT&T, SKT team on 'xRAN' to bring SDN, NFV to the RAN

by *Anne Morris* | Oct 12, 2016 4:00pm

Source: https://www.fiercewireless.com/europe /deutsche-telekom-touts-benefits-software-based-ran

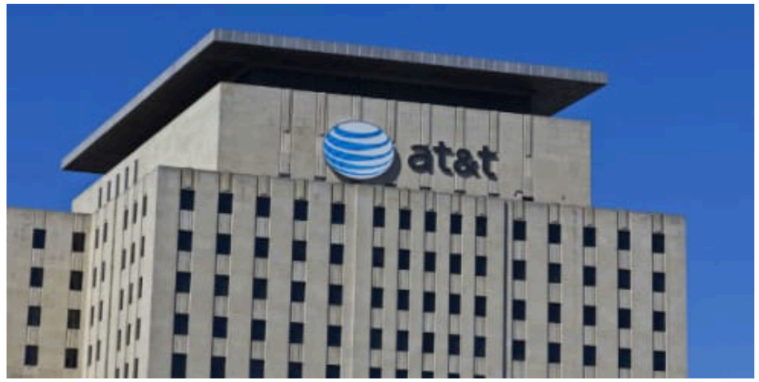
Facebook, Google use SDN to boost data center connectivity

by
Lee Doyle
Doyle Research

Internet content providers like Facebook and Google are already using SDN to improve hyperscale data center connectivity. But more could be done.

Source: http://searchsdn.techtarget.com/tip/Facebook-Google-use-SDN-to-boost-data-center-connectivity

AT&T to Join Verizon in Working With KT on NFV, SDN, and 5G



Dan Meyer
June 15, 2017
10:44 am PT

AT&T is set to join Verizon in working with South Korean operator KT on software-defined networking (SDN), network functions virtualization (NFV), and 5G.

Source: https://www.sdxcentral.com/articles/news/ att-to-join-verizon-in-working-with-kt-on-nfv-sdn-and-5g/2017/06/

SDN adoption: Military

DEFENSE

Pentagon considering push to software-defined networking

Source: https://www.fedscoop.com/pentagon-considering-push-software-defined-networking/

New Technology Approaches Can Solve Complex U.S. Navy Problems

February 1, 2017

By Cmdr. Jamie Gateau, USN (Ret.)

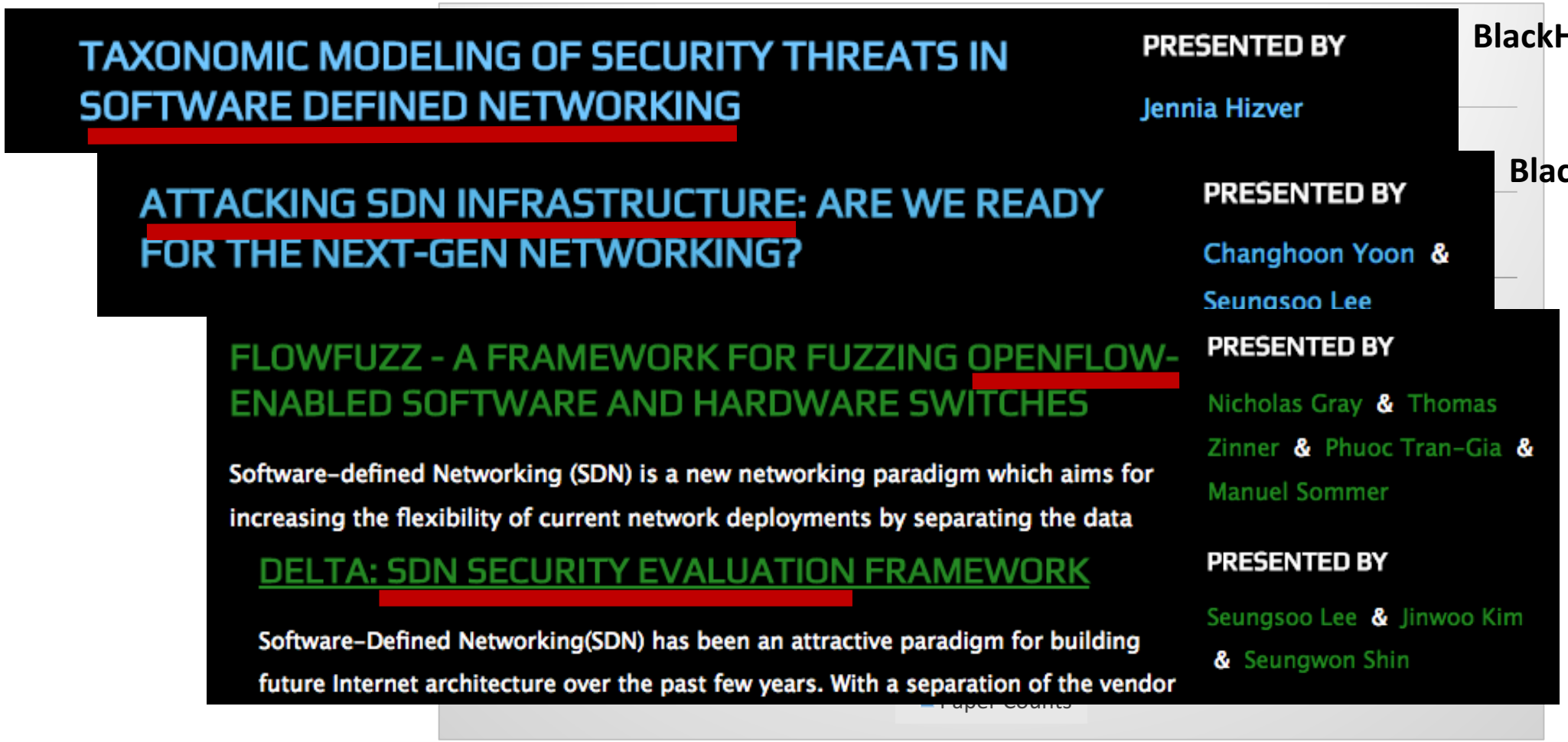
Software-defined networking, virtualization and orchestration enable [illegible] security, capability and flexibility ashore.

U.S. Navy commanders often struggle to deliver uninterrupted communications at sea without the added complications of providing command and control in denied or degraded environments. They face a double whammy of operational and technical hurdles.

Source: https://www.afcea.org/content/Article-new-technology-approaches-can-solve-complex-us-navy-problems

*But, what about **SECURITY**?*

Attention to SECURITY has been growing!



BlackHat USA 15' Briefing

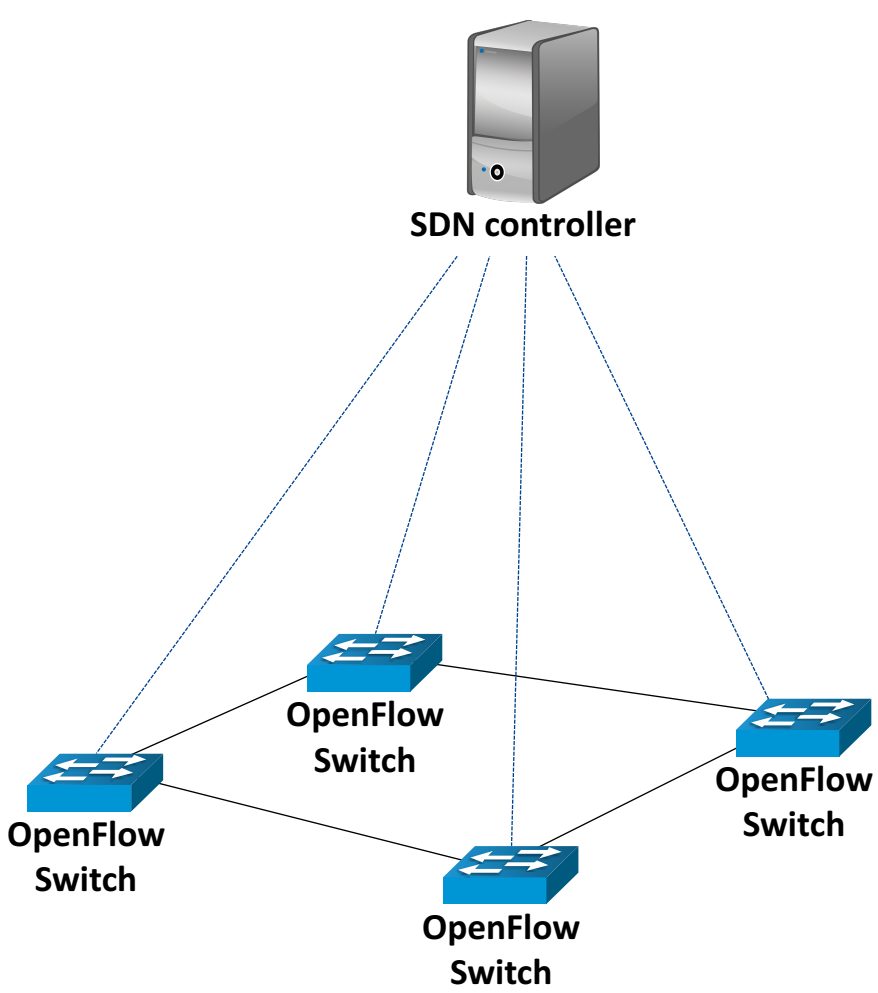
BlackHat USA 16' Briefing

BlackHat USA 17' Briefing

BlackHat USA 17' Arsenal

** Google scholar [scholar.google.com]*

Attack Vectors in SDN architecture

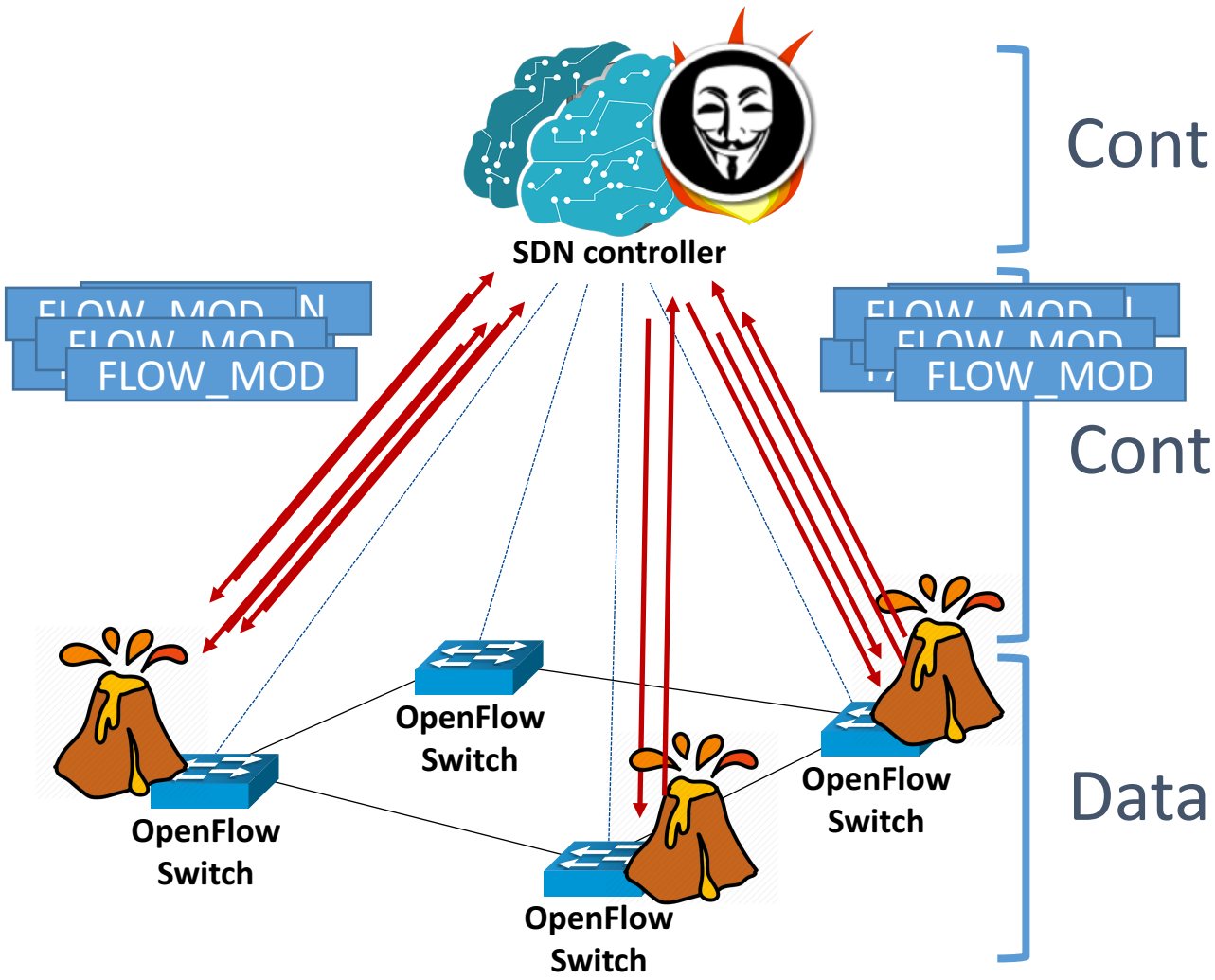


Control plane

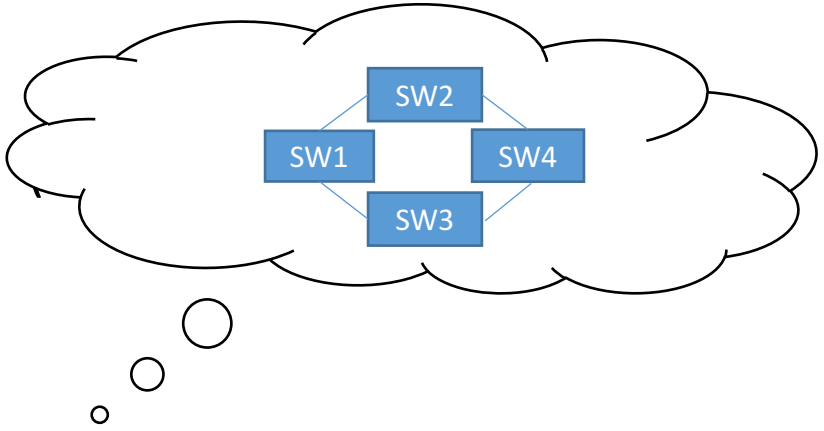
Control channel

Data plane

Attack Examples



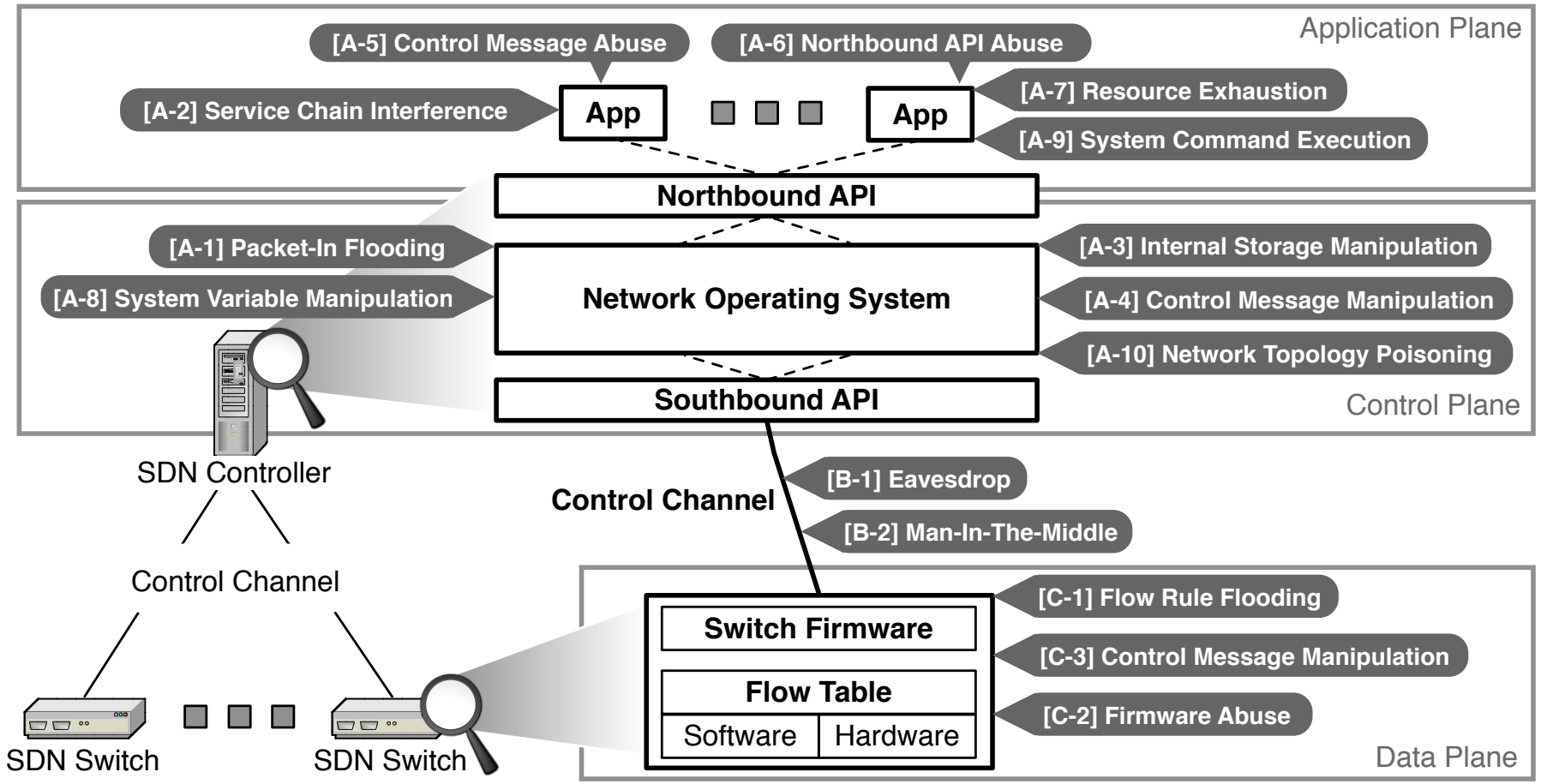
Control plane



Control channel (e.g., Eavesdropping attack)

Data p ing attack)

SDN Vulnerability Genome Project [1]



Control plane

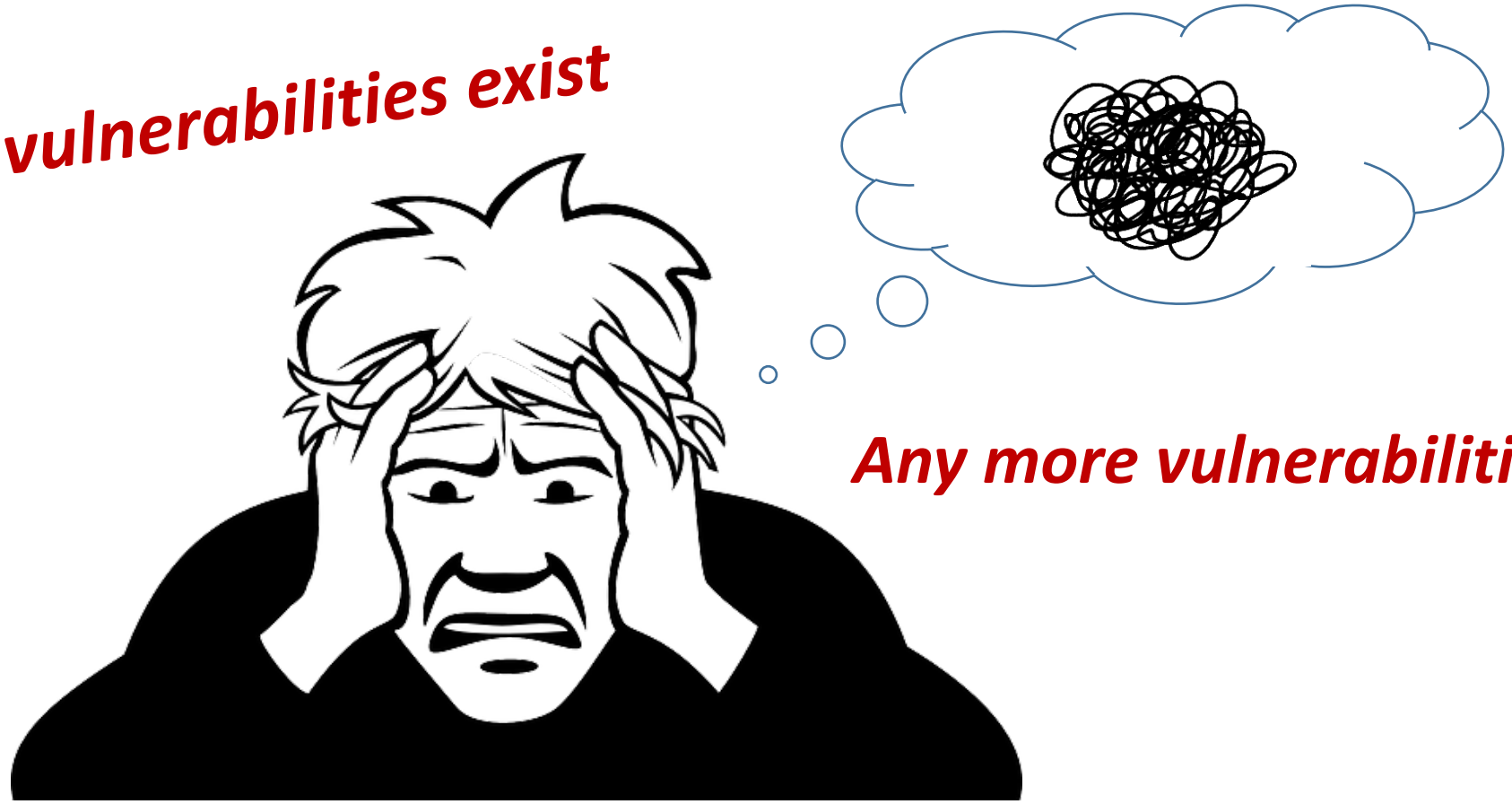
Control channel

Data plane

[1] Yoon, Changhoon, et al. "Flow wars: Systemizing the attack surface and defenses in software-defined networks." *IEEE/ACM Transactions on Networking* 6 (2017): 3514-3530.

Network admin's concerns…

How many vulnerabilities exist now?



Any more vulnerabilities?

How to reproduce each test case?

Don’t worry, run DELTA

- **DELTA: A Security Assessment Framework for SDN**

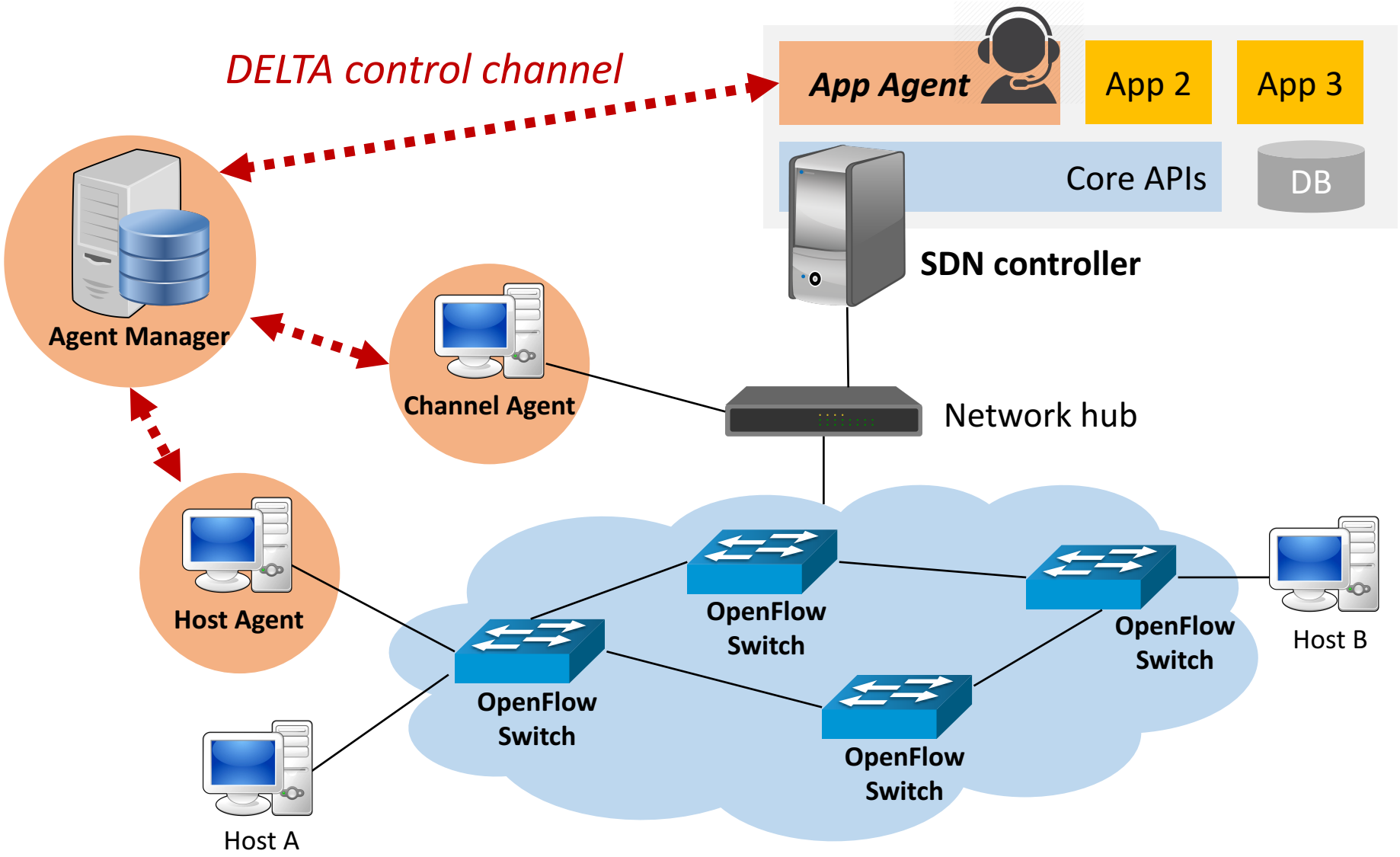
Automating a working process

Finding new attacks

Supporting diverse SDN components

Covering many attack cases

DELTA: System Design



DELTA: System Design

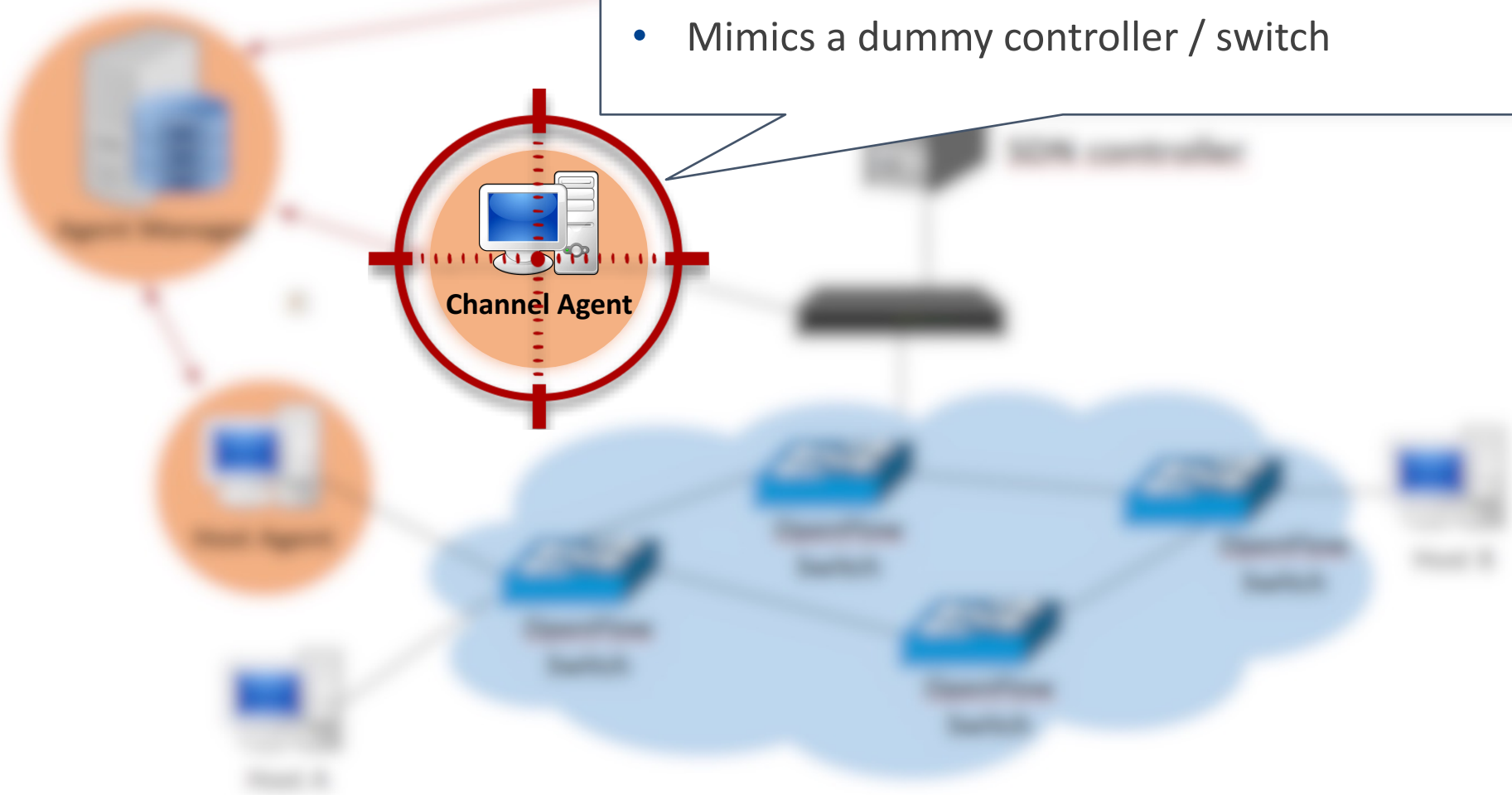
- **Agent Manager**
- The '**Control tower**'
- Remotely controls the agents deployed to the target network
- Leverages different agents to perform various security test cases
- Analyzes the test results collected from the agents

DELTA: System Design

- **Application Agent**
- SDN applications that conduct attack procedures as instructed by the manager
- Implements the known **malicious functions** as an application agent library
- Includes **fuzzing modules** that randomize the SDN control flows

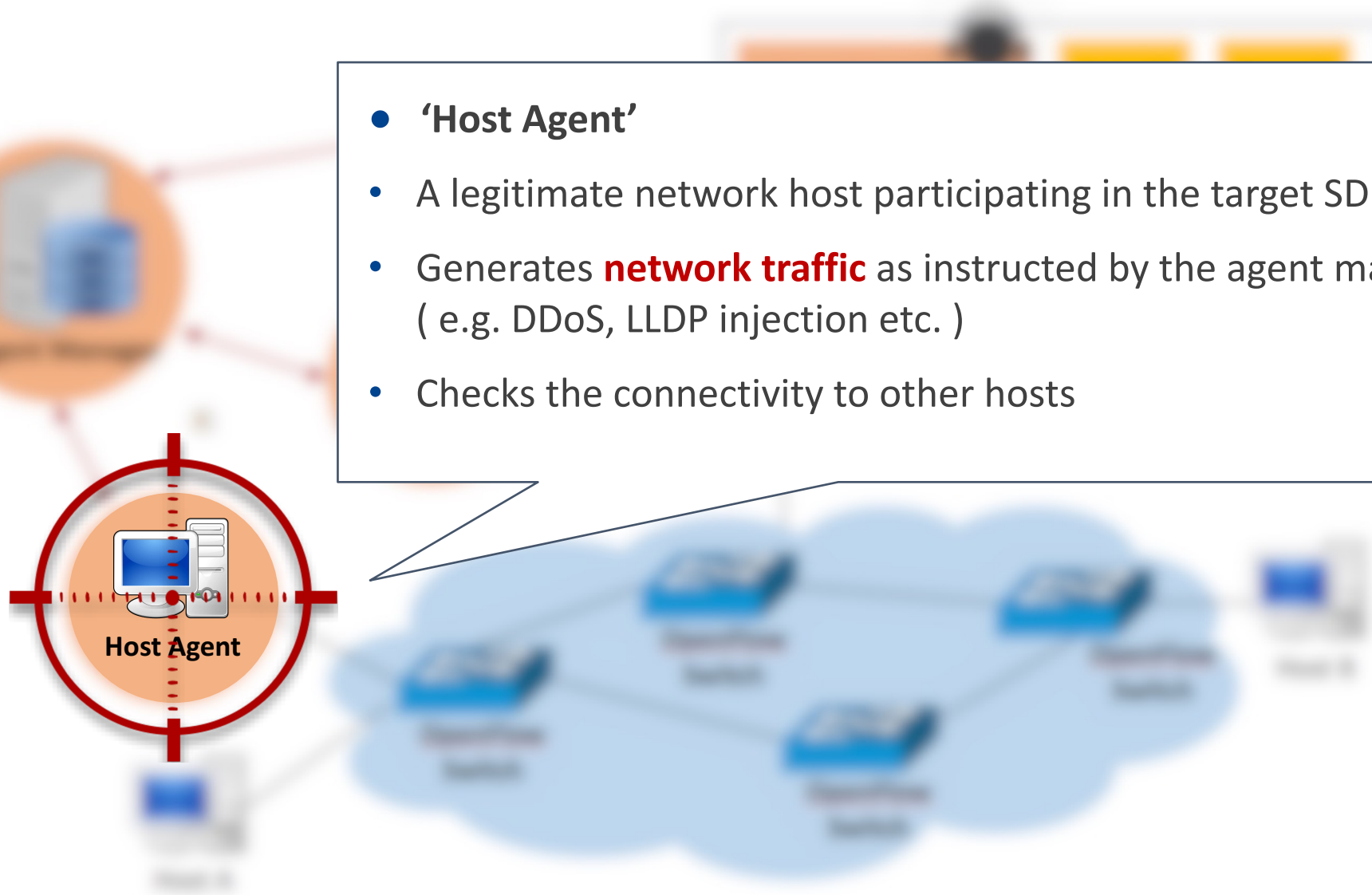
DELTA: Sys

- **'Channel Agent'**
- Located between the controller and the switch
- Includes **fuzzing modules** that sniff and modify the unencrypted **SDN control messages**
- Mimics a dummy controller / switch



DELTA: System Design

- **'Host Agent'**
- A legitimate network host participating in the target SDN
- Generates **network traffic** as instructed by the agent manager (e.g. DDoS, LLDP injection etc.)
- Checks the connectivity to other hosts



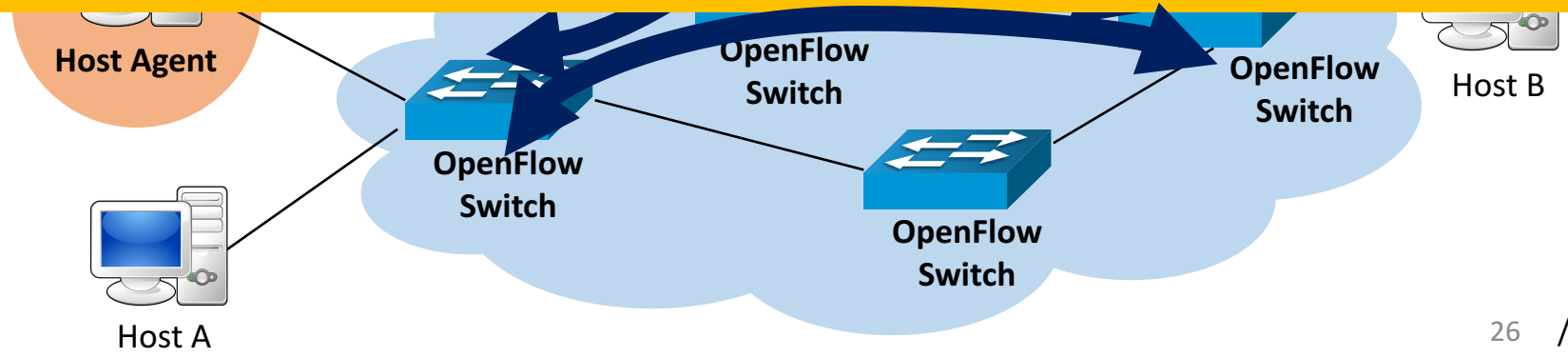
Automated Operation

Automating a working process

2. **Instruct** each agent to conduct the test

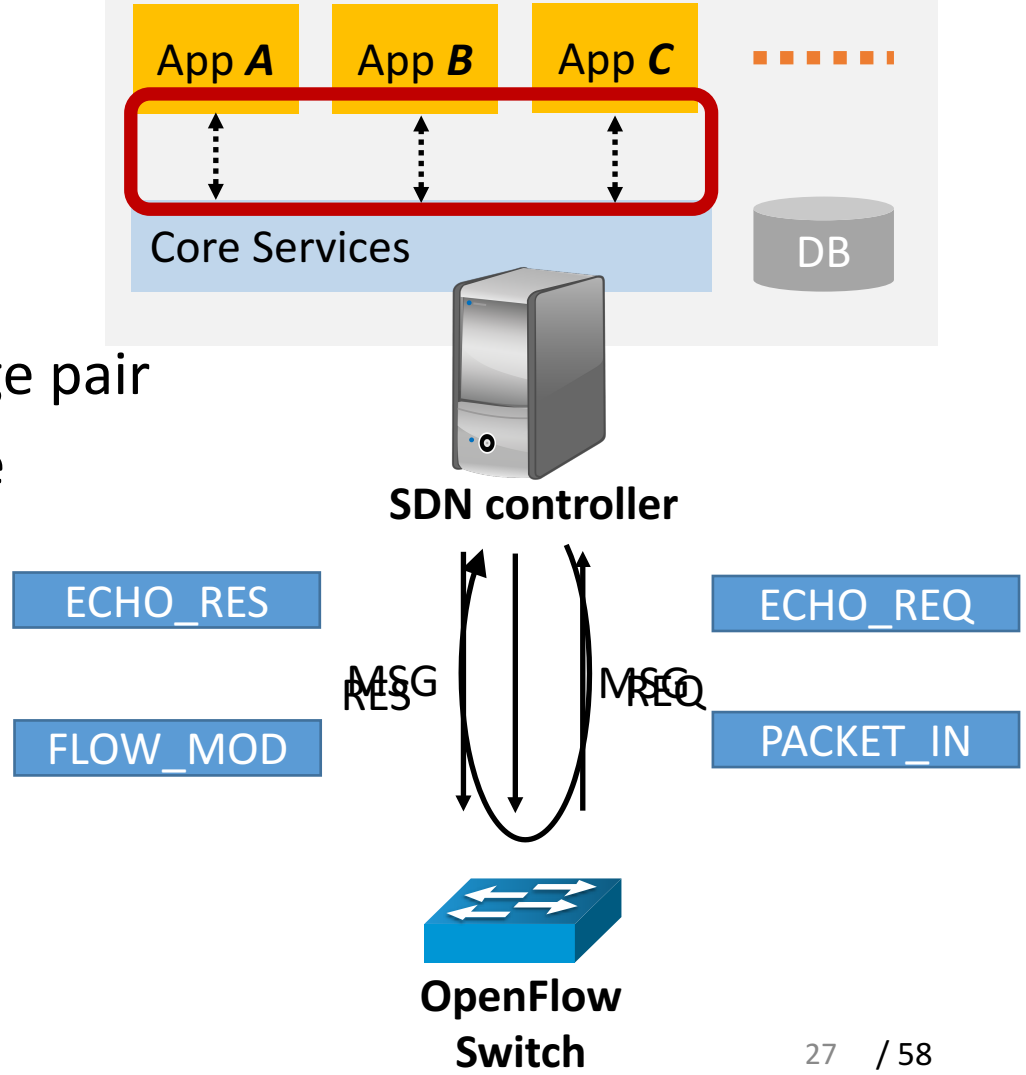
3. **Collect** the result of the test from each agent

4. **Notify** the result



SDN Control Flow Fuzzing

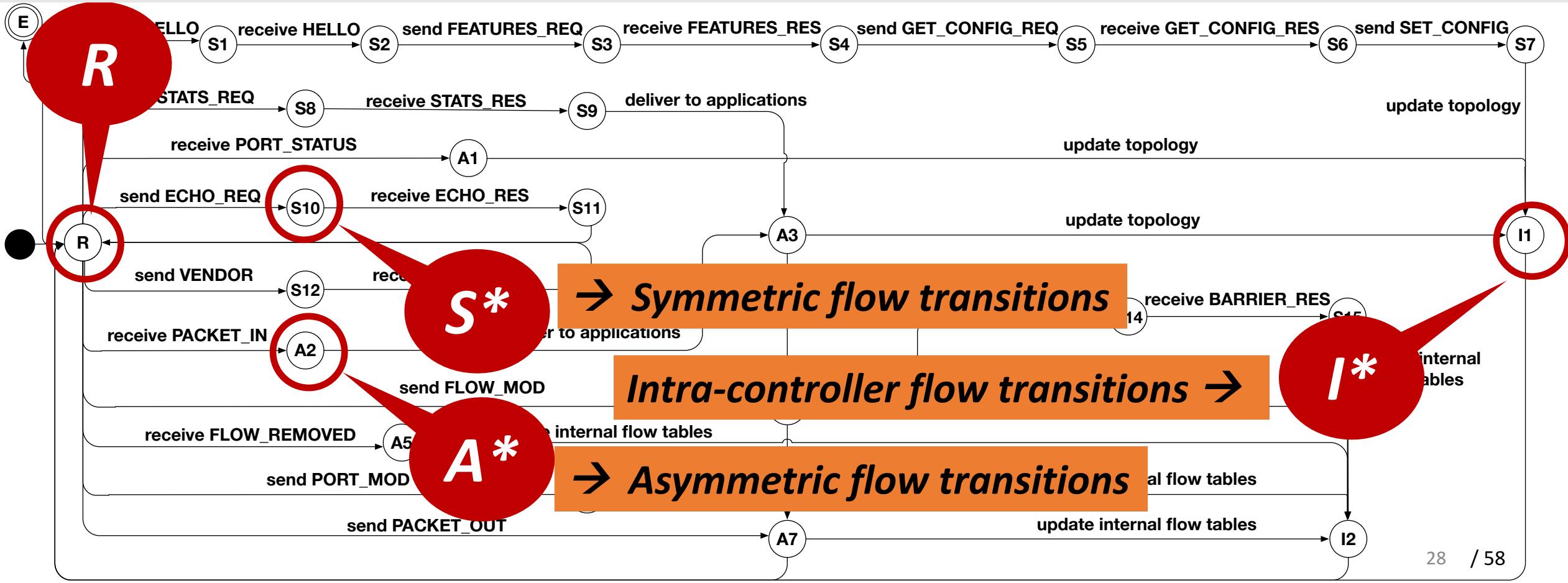
- Find **NEW** security holes in SDN (i.e., OpenFlow protocol based)
- Define three types of control flow operations
  1. **Symmetric** control flow: Req. & Res. message pair
  2. **Asymmetric** control flow: One-way message
  3. **Intra-controller** control flow: between applications and core services



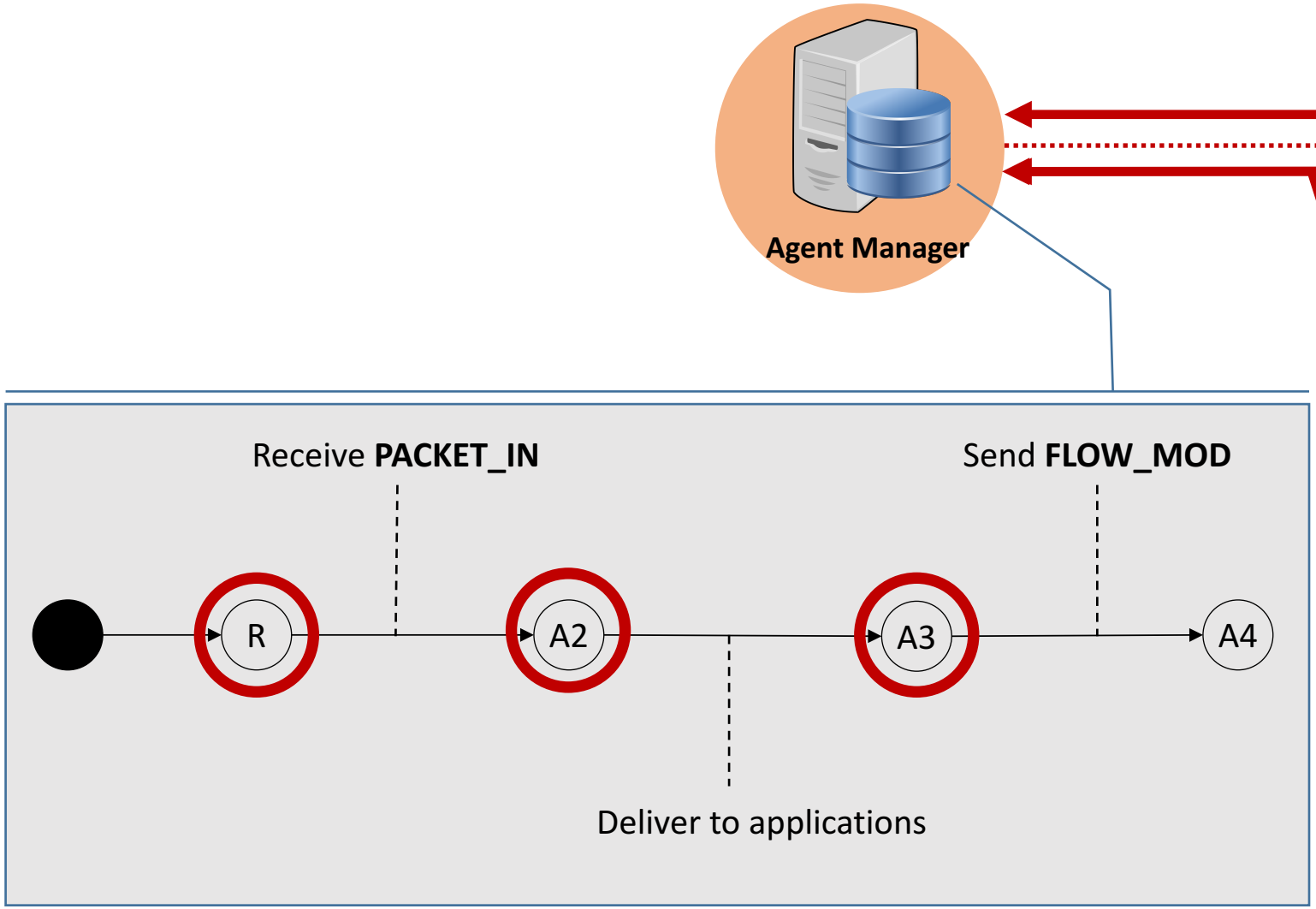
Operational State Diagram

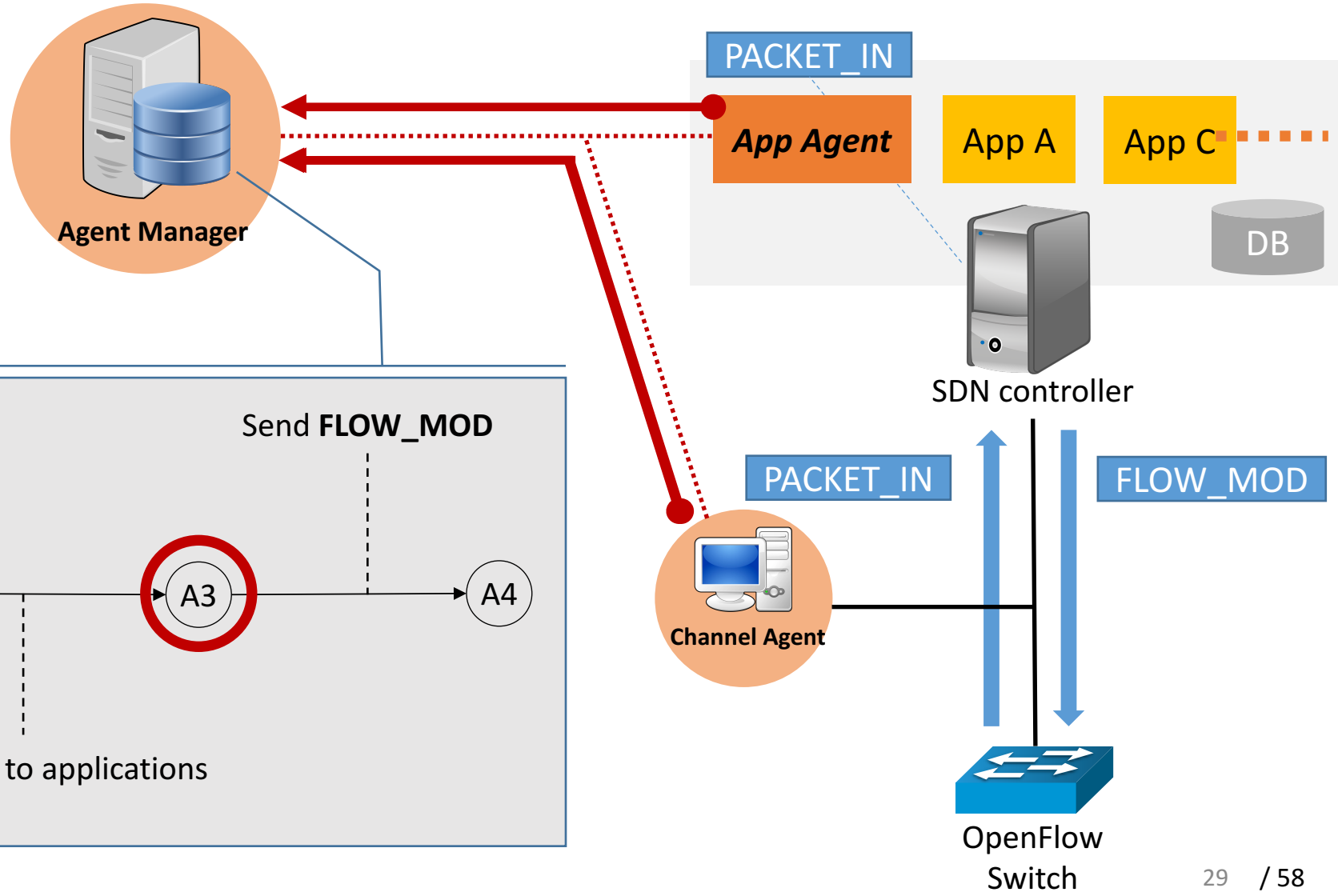
To find new vulnerabilities,

1. ***Infer*** *the current state of the controller*
2. ***Manipulate*** *the control flow sequence or the input values*

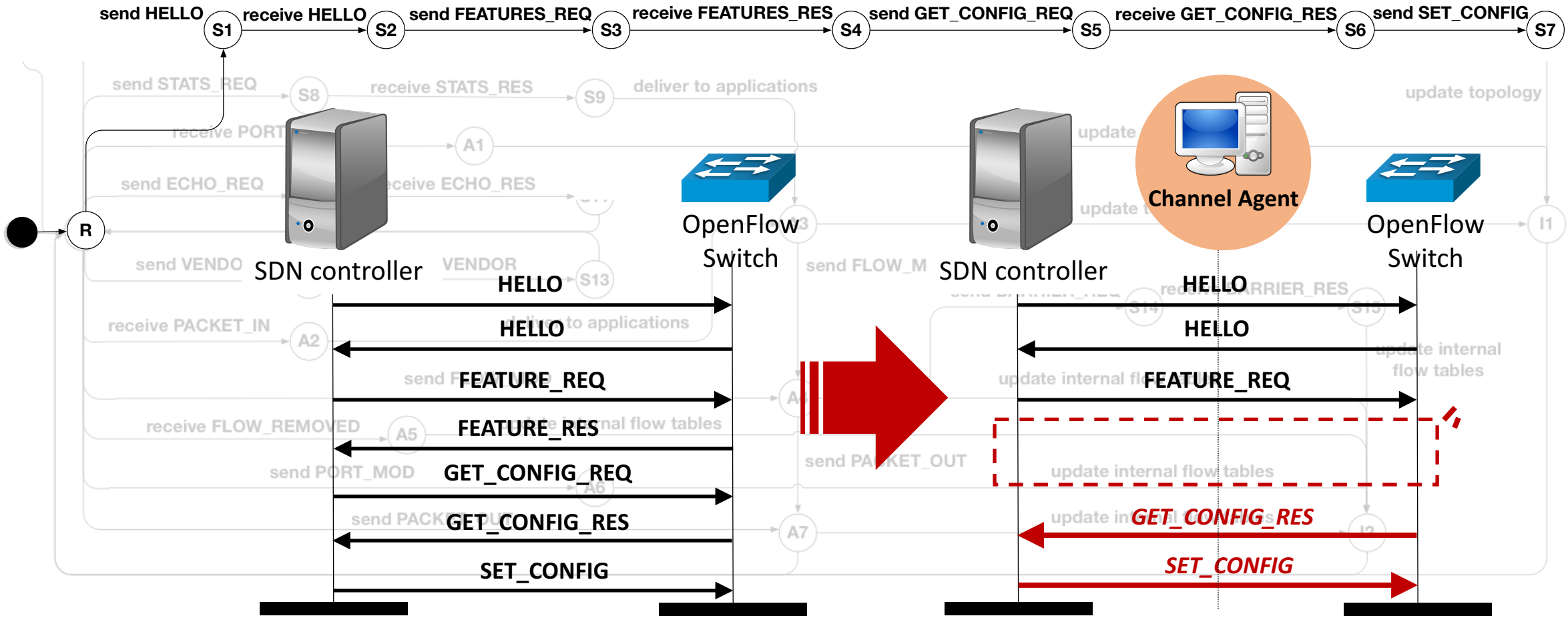


Identifying Current State of Controller

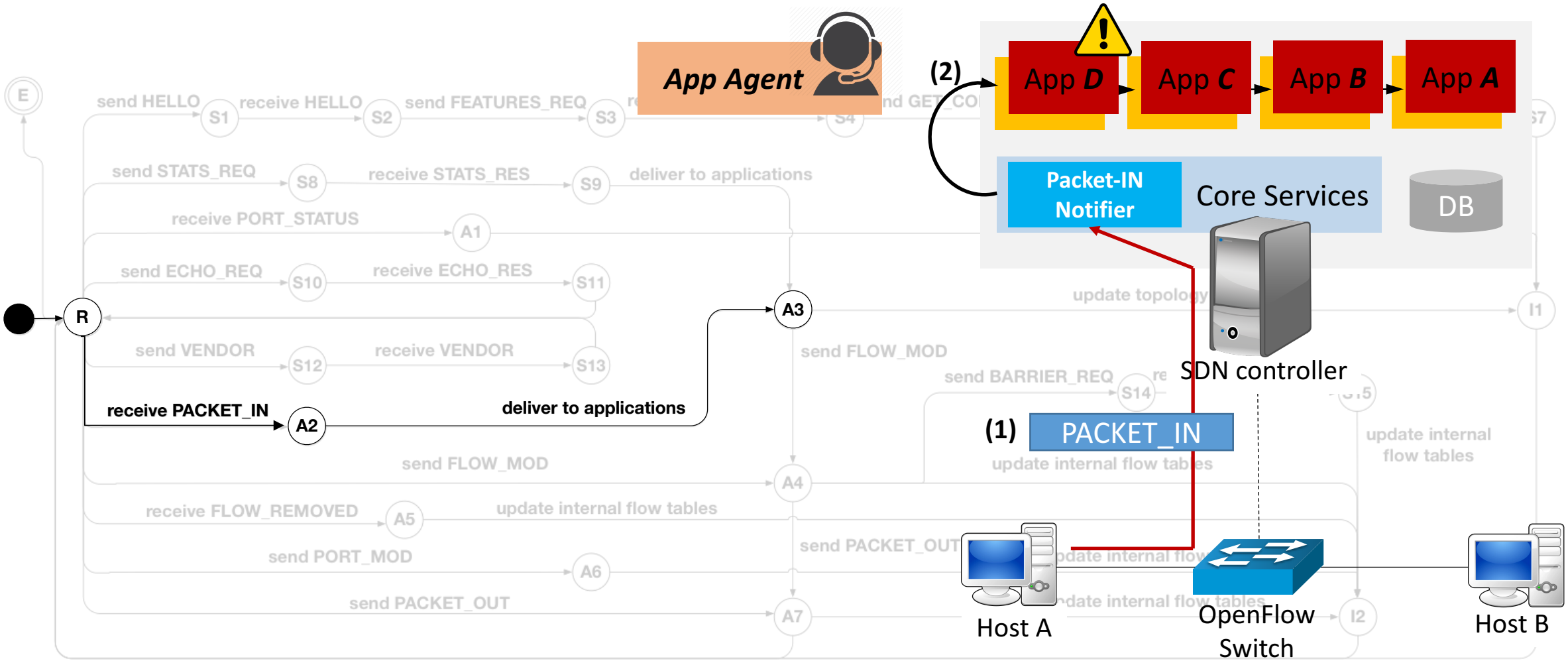




Randomizing Symmetric Control Flow Sequence

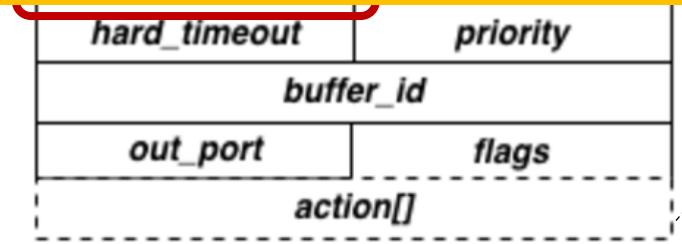


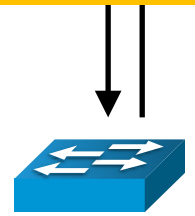
Randomizing Asymmetric Control Flow Sequence



Randomizing Input Values

Finding new attacks





e.g.) ADD (0x0000) → ***(Undefined) (0xFFFF)***

Implementation

- Program languages: Java / Python

```
[INFO] Reactor Summary:
[INFO]
[INFO] delta-manager ..................................... SUCCESS [ 13.459 s]
[INFO] delta-agent-apps .................................. SUCCESS [  0.005 s]
[INFO] delta-agent-app-onos-1.13.1 ....................... SUCCESS [  2.111 s]
[INFO] delta-agent-apps-floodlight ....................... SUCCESS [  0.003 s]
[INFO] delta-agent-app-floodlight-1.2 .................... SUCCESS [ 14.281 s]
[INFO] delta-agent-apps-opendaylight ..................... SUCCESS [  0.003 s]
[INFO] delta-agent-app-odl-oxygen-sr2 .................... SUCCESS [  1.558 s]
[INFO] delta-agent-channel ............................... SUCCESS [  8.409 s]
[INFO] delta-agent-host .................................. SUCCESS [  0.218 s]
[INFO]
[INFO] BUILD SUCCESS
```

[LOC]

4537

721

821

808

3835

213

Supported SDN Components

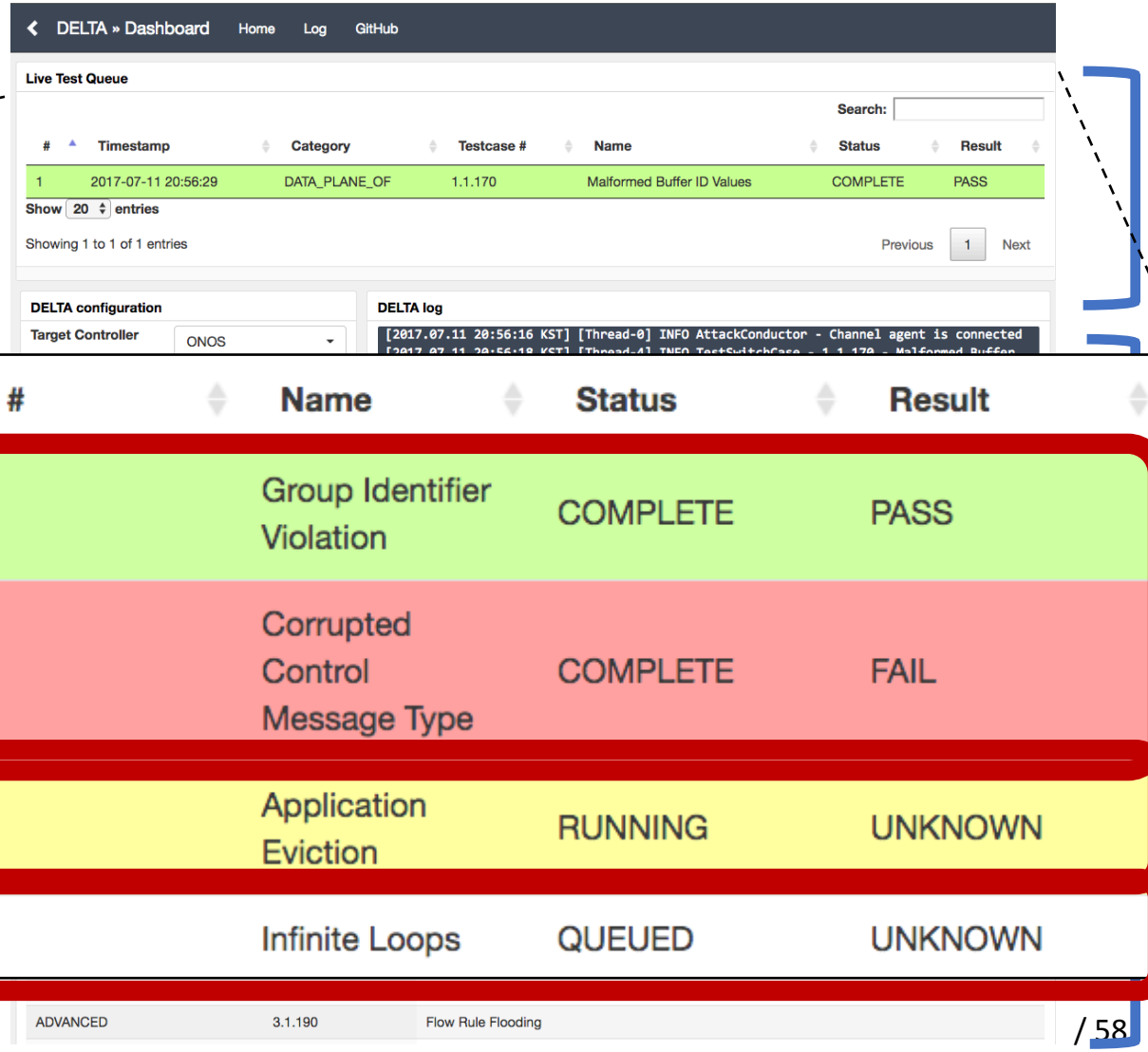
- Supports four different SDN controllers

Supporting diverse SDN components

Pica8	P-3290	1.0, 1.3
Arista Networks	7050-T36	1.0
HPE	E3800 24G-2SFP+	1.0
Linux Foundation Collaborative Project	OpenVSwitch	1.0, 1.3

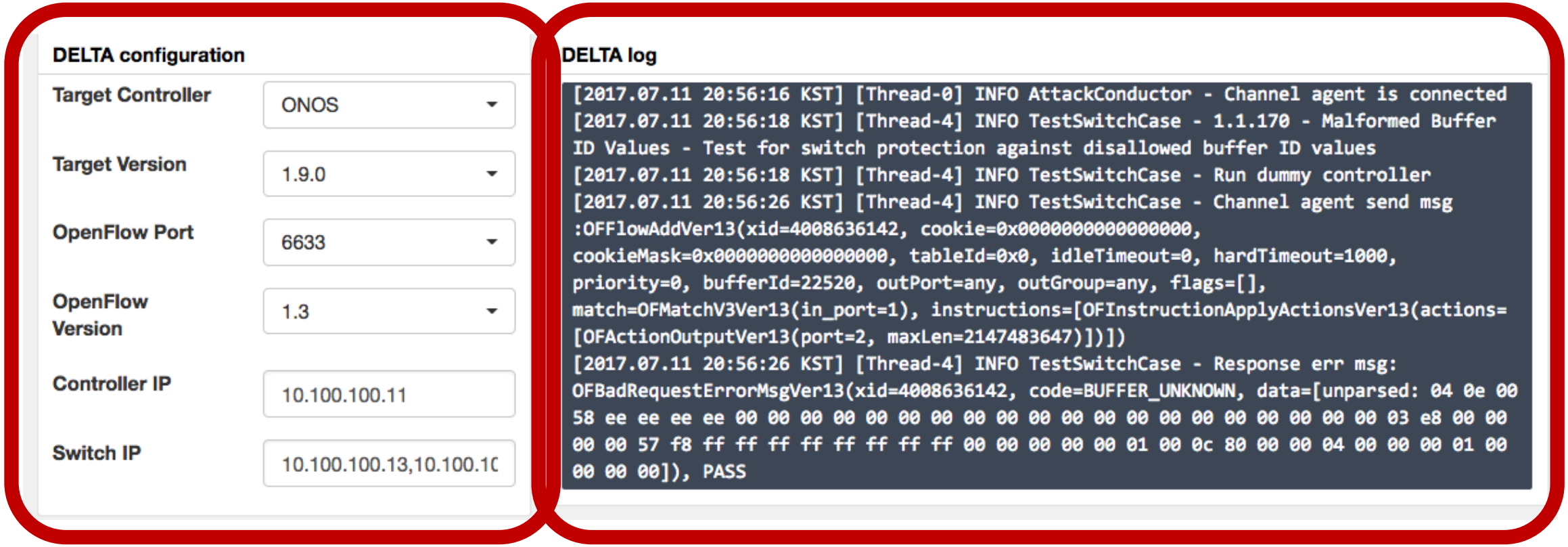
Web-based UI

Live test queue:



#	Timestamp	Category	Testcase #	Name	Status	Result
1	2016-09-12 08:56:26	DATA_PLANE_OF	1.1.030	Group Identifier Violation	COMPLETE	PASS
2	2016-09-12 08:56:57	CONTROL_PLANE_OF	2.1.020	Corrupted Control Message Type	COMPLETE	FAIL
3	2016-09-12 08:57:15	ADVANCED	3.1.100	Application Eviction	RUNNING	UNKNOWN
4	2016-09-12 08:57:14	ADVANCED	3.1.030	Infinite Loops	QUEUED	UNKNOWN

Configuration and Log Pane



Test Case Inventory

DELTA Test Cases

Select all | Deselect all | Run Selected Entries

Category	Testcase #	Name
ADVANCED	3.1.100	Application Eviction
ADVANCED	3.1.110	Memory Exhaustion
ADVANCED	3.1.120	CPU Exhaustion
ADVANCED	3.1.130	System Variable Manipulation
ADVANCED	3.1.140	System Command Execution

- **Test set 1: Data plane security**

40+

Covering many attack cases

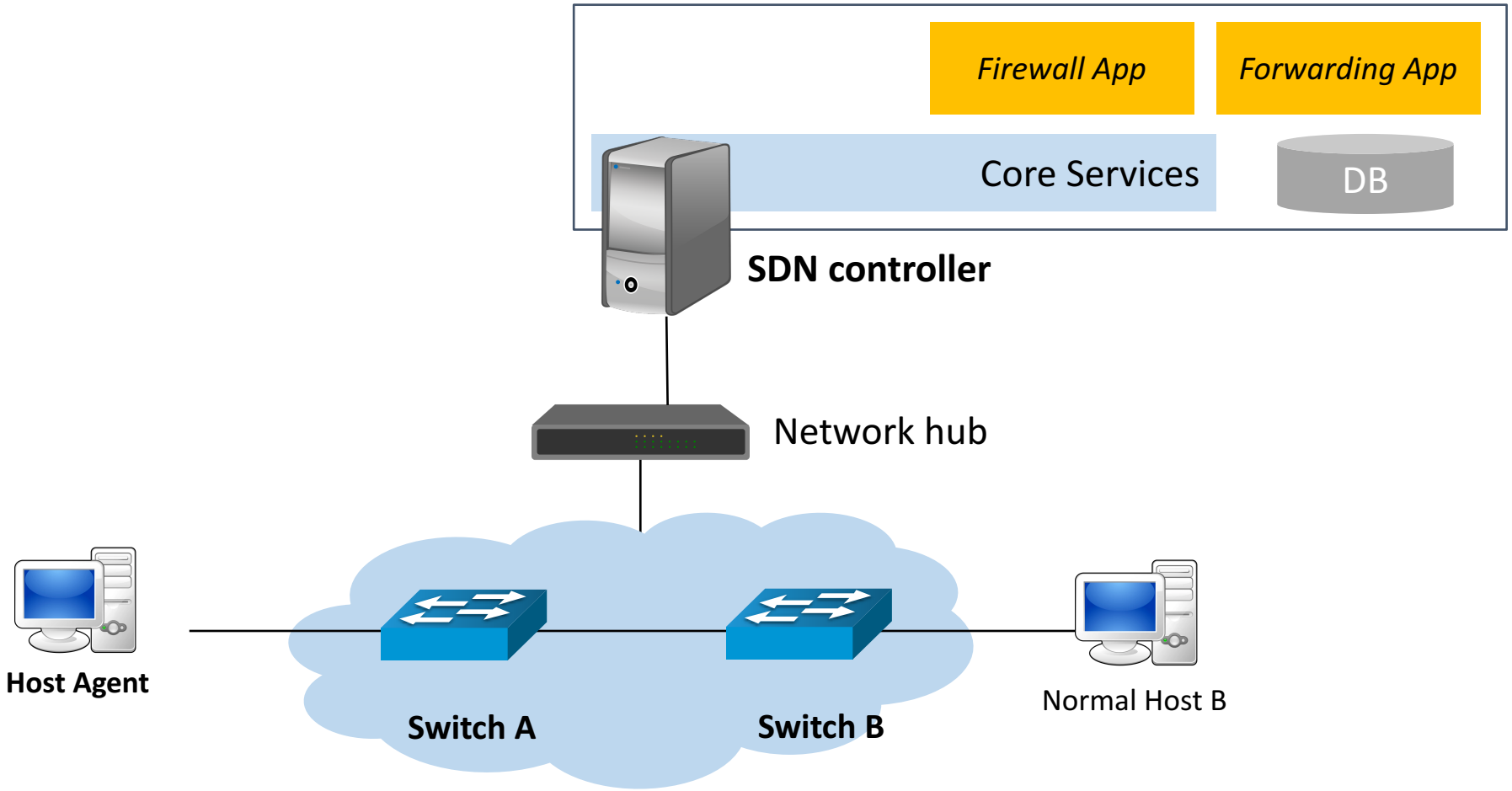
- e.g., SDN applications exploiting SDN controllers' architectural vulnerabilities

Let's start DEMO time!

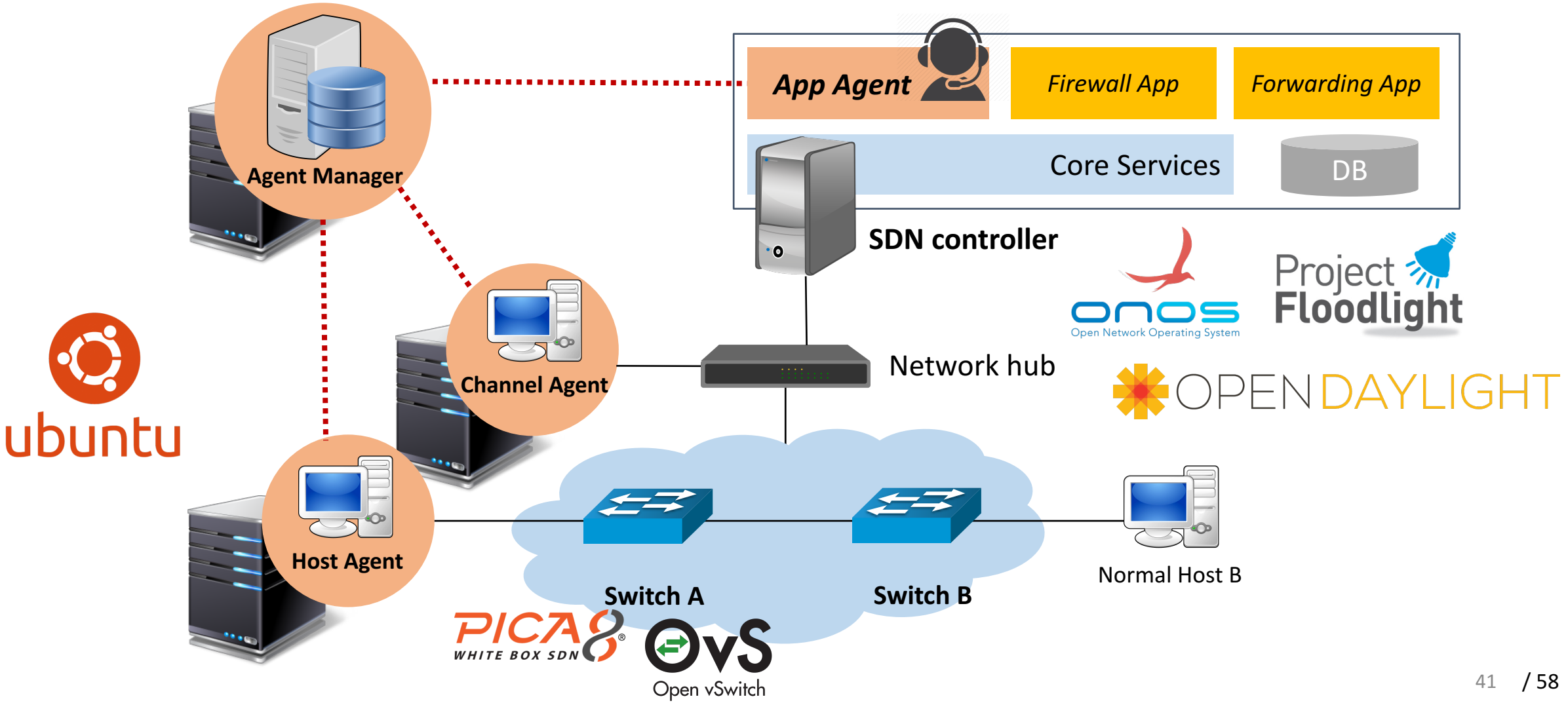
Demonstration

- Test environments
- 1 KNOWN attack for Floodlight
- 2 NEW attacks for ONOS, OpenDaylight

Test Environments

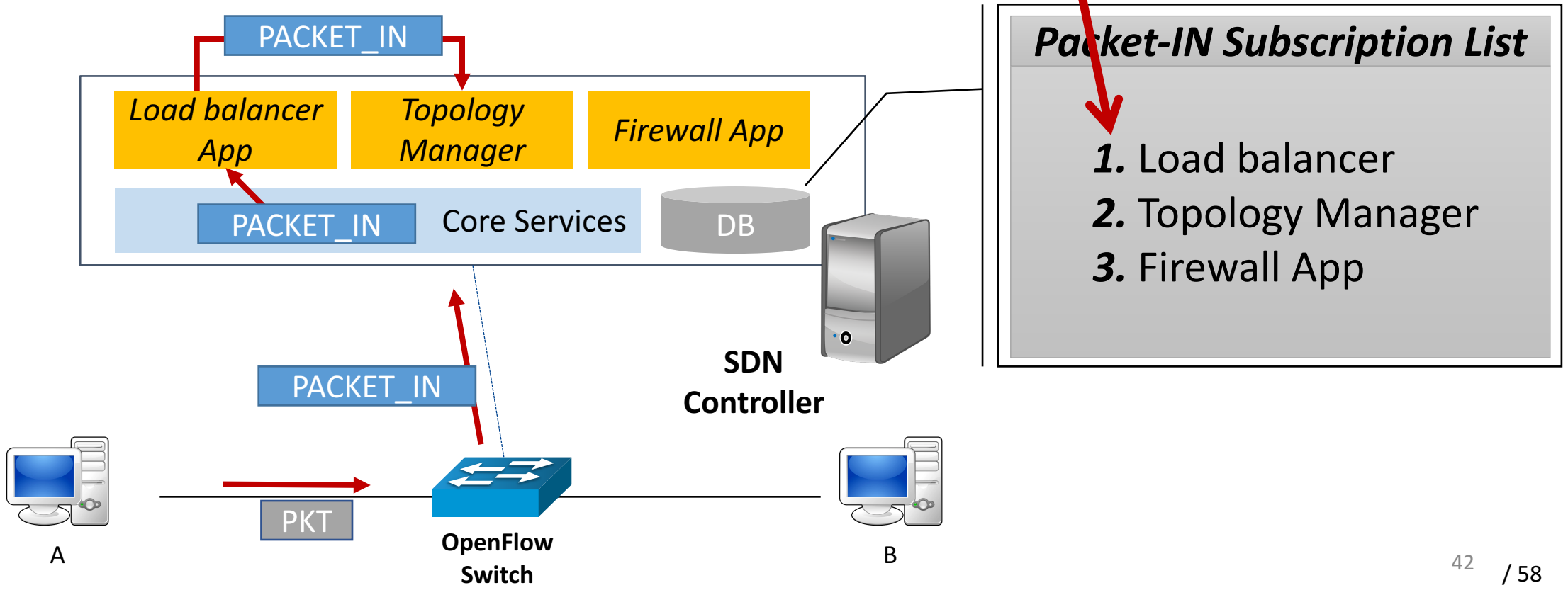


Test Environments

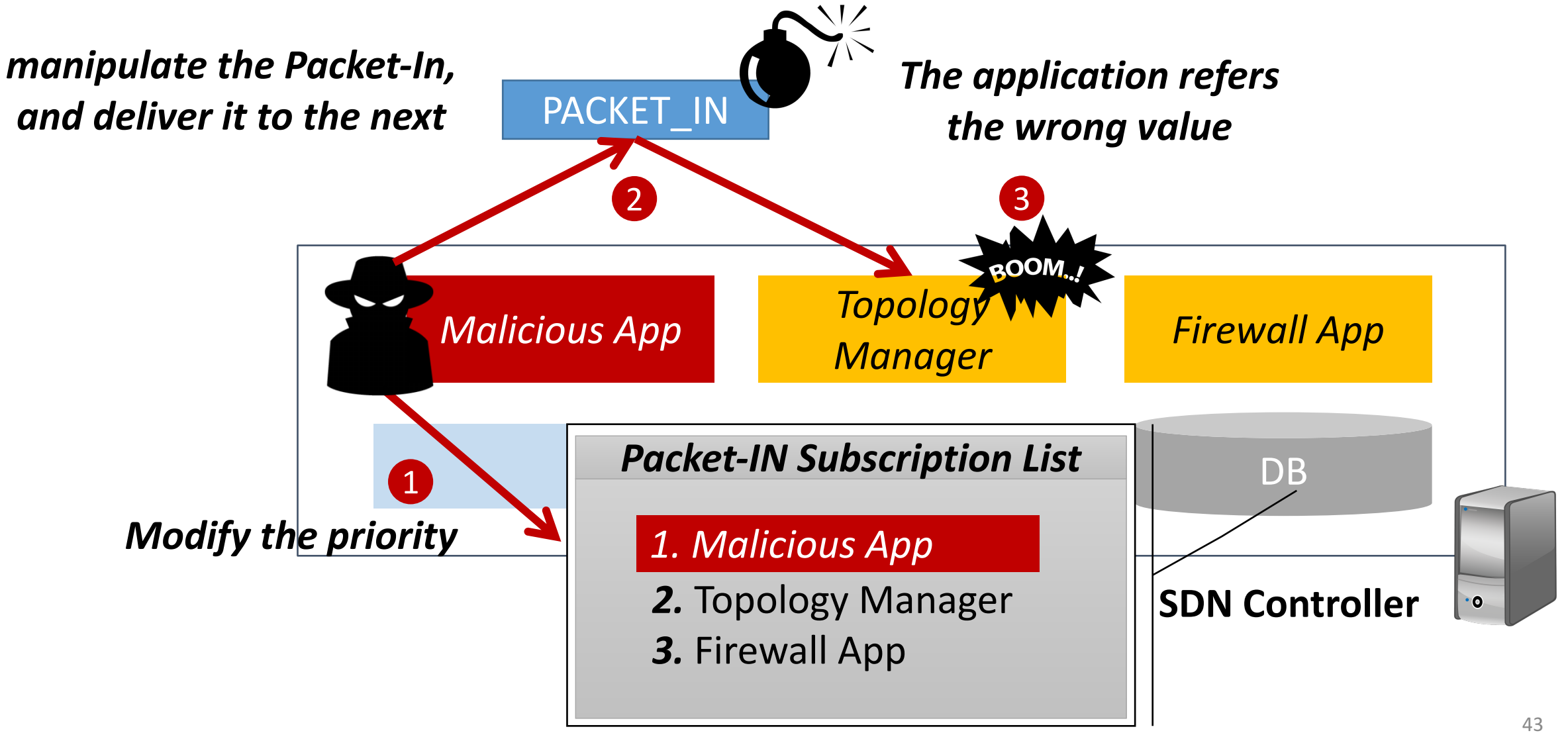


Event Subscription in SDN

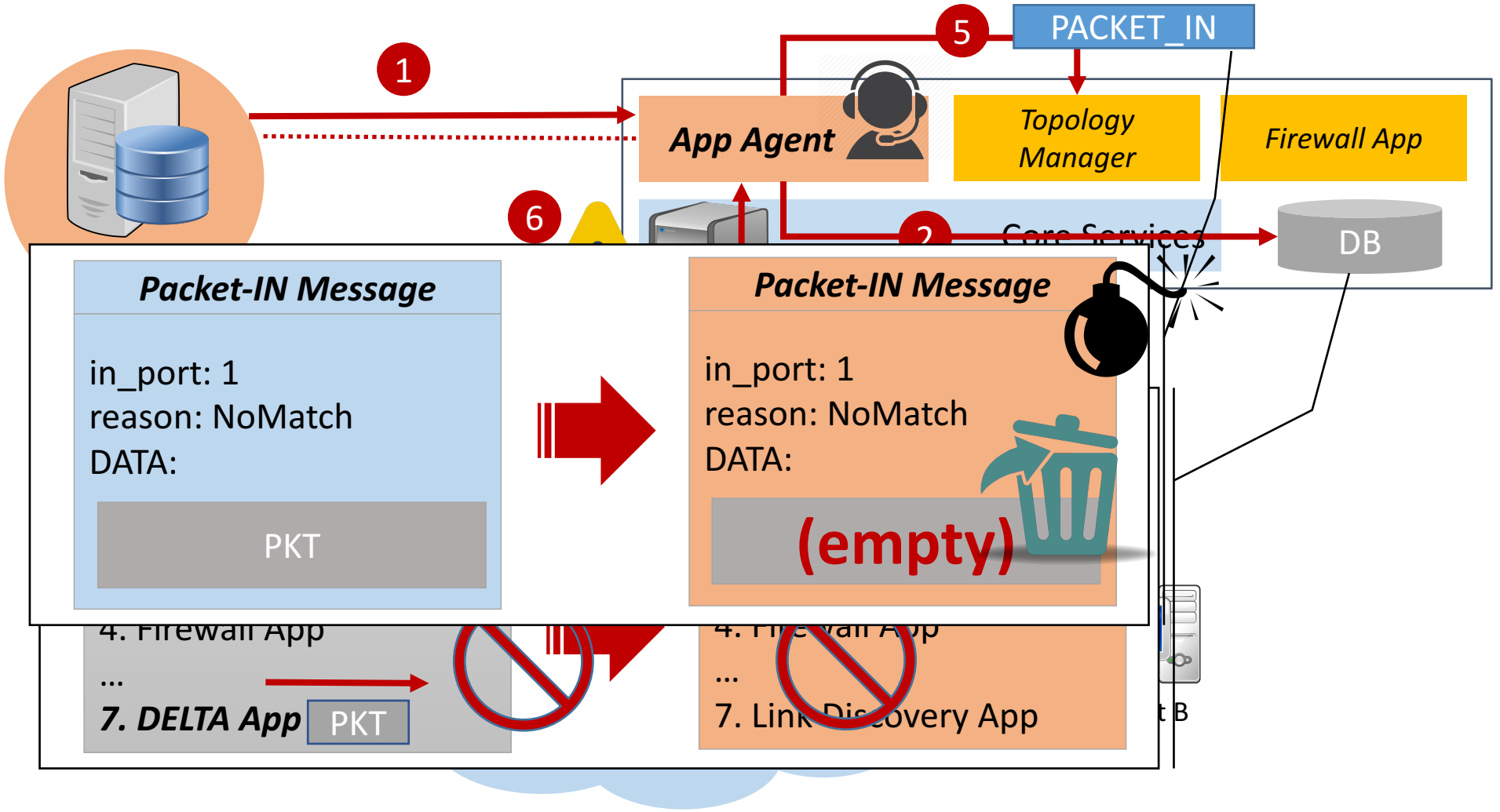
- An SDN controller maintains ***an event subscription list***
- Packet-In events are processed according to ***a priority***



Attack Strategy: Smash the subscription!

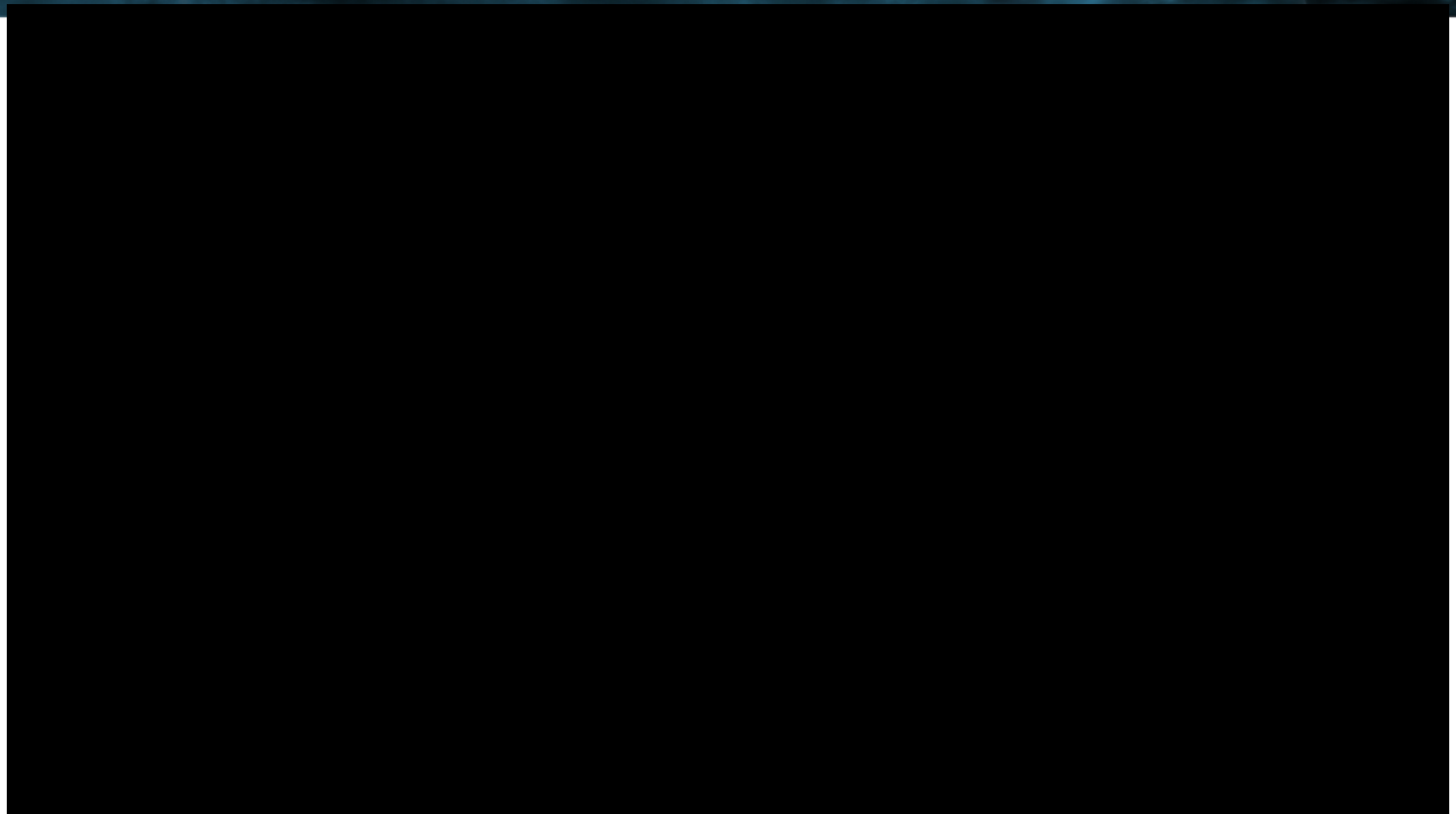


DEMO 1: Packet-In Data Forge attack



6 The [illegible] the packet-in message to the [illegible] and then passes it over to the next one in the subscription list

DEMO 1: Packet-In Data Forge attack



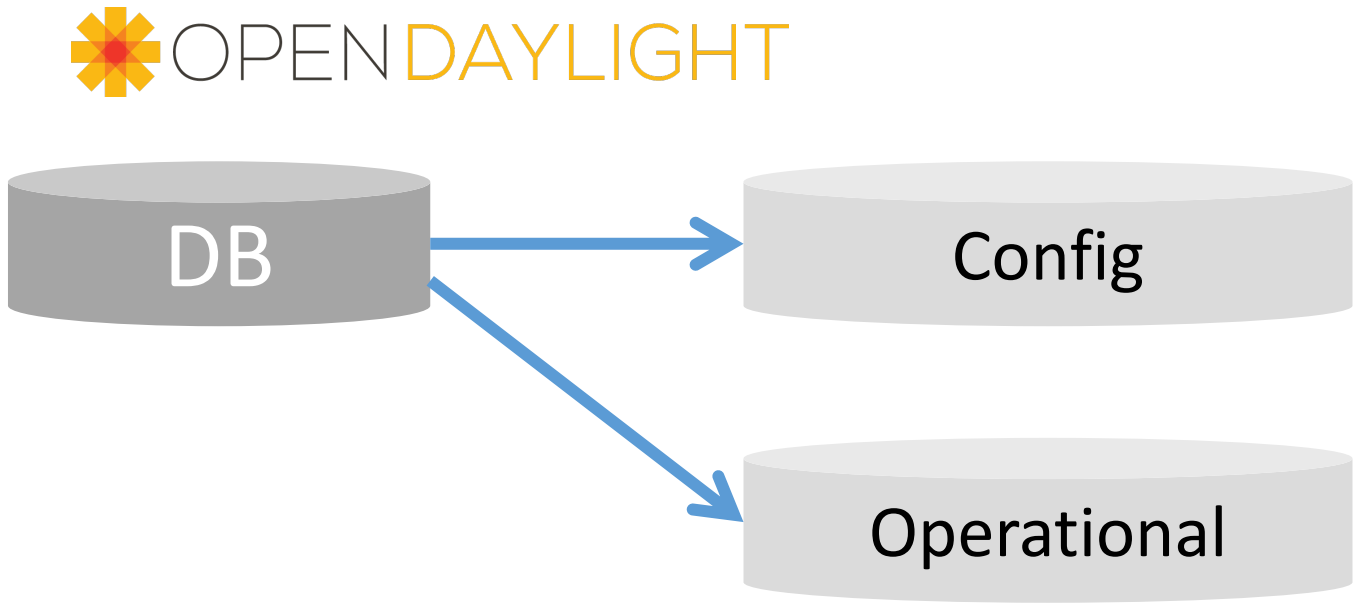
DEMO 1: Packet-In Data Forge attack

- Feasible to Floodlight 1.1
- ***Why?***
  - SDN applications granted ***powerful authority***
- ***How to defend?***
  - Policy-based ***access control*** to SDN applications
    - e.g., Security-Mode ONOS [1]

[1] https://wiki.onosproject.org/display/ONOS/Security-Mode+ONOS

Databases in OpenDaylight

- OpenDaylight (ODL) manages two types of databases

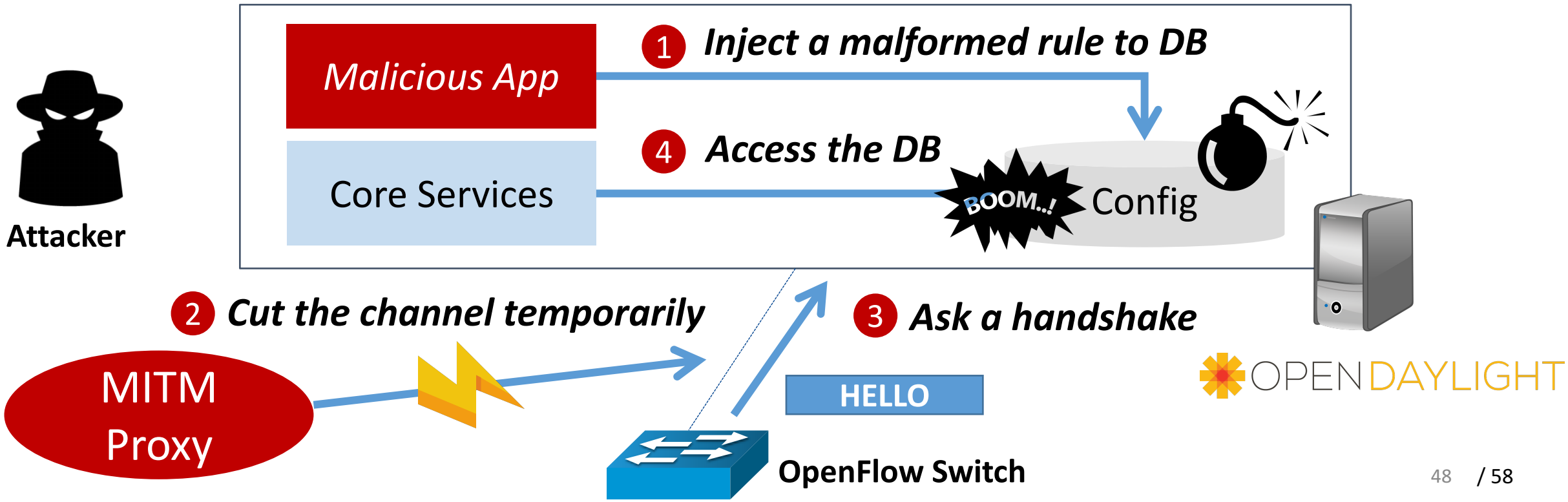


*Proactive and **persistent** rules,*
***Non-volatile** memory*

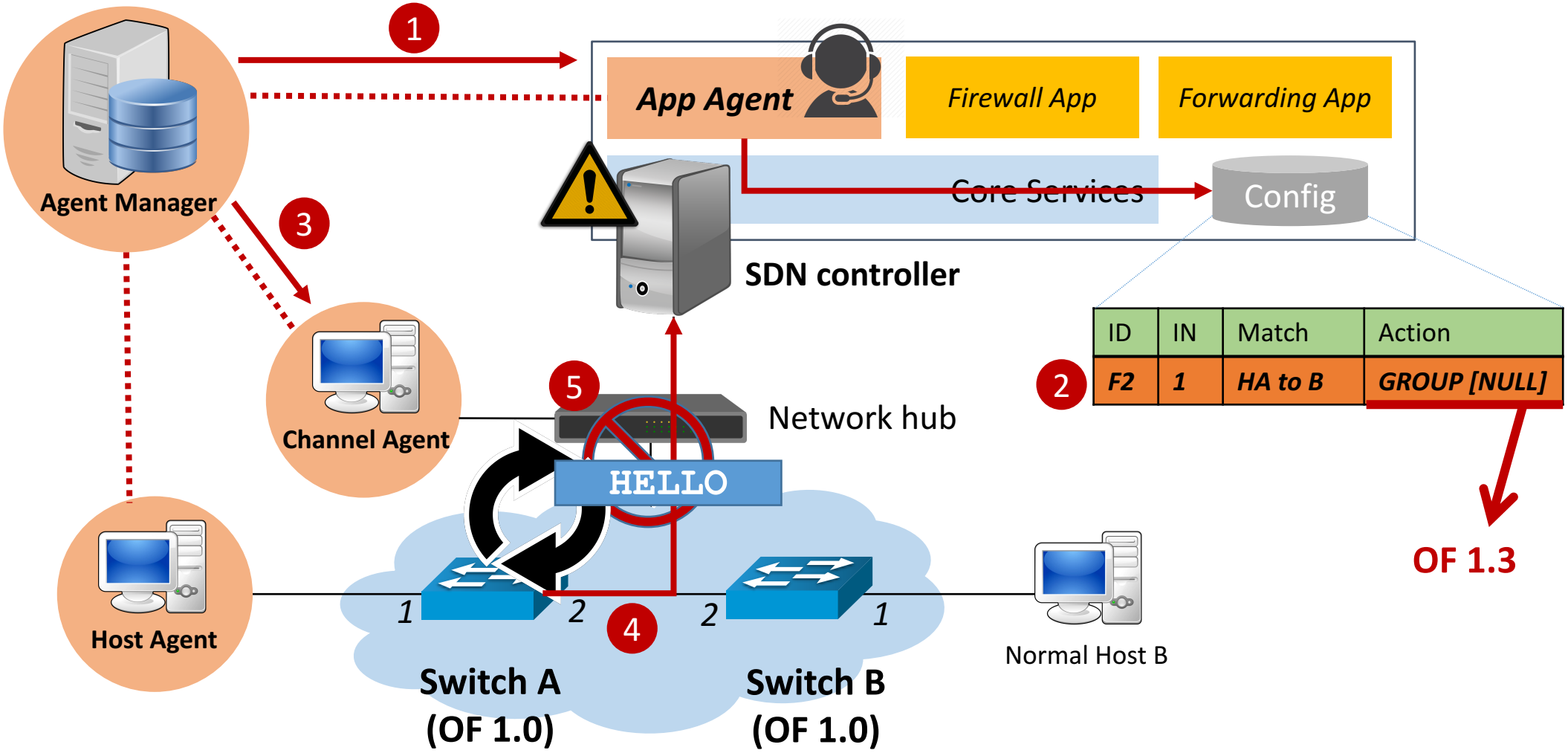
Reactive and temporary rules
Volatile memory

Attack Strategy: Exploit the config. DB

- ODL ***refers*** the configuration DB, when handshaking with a switch

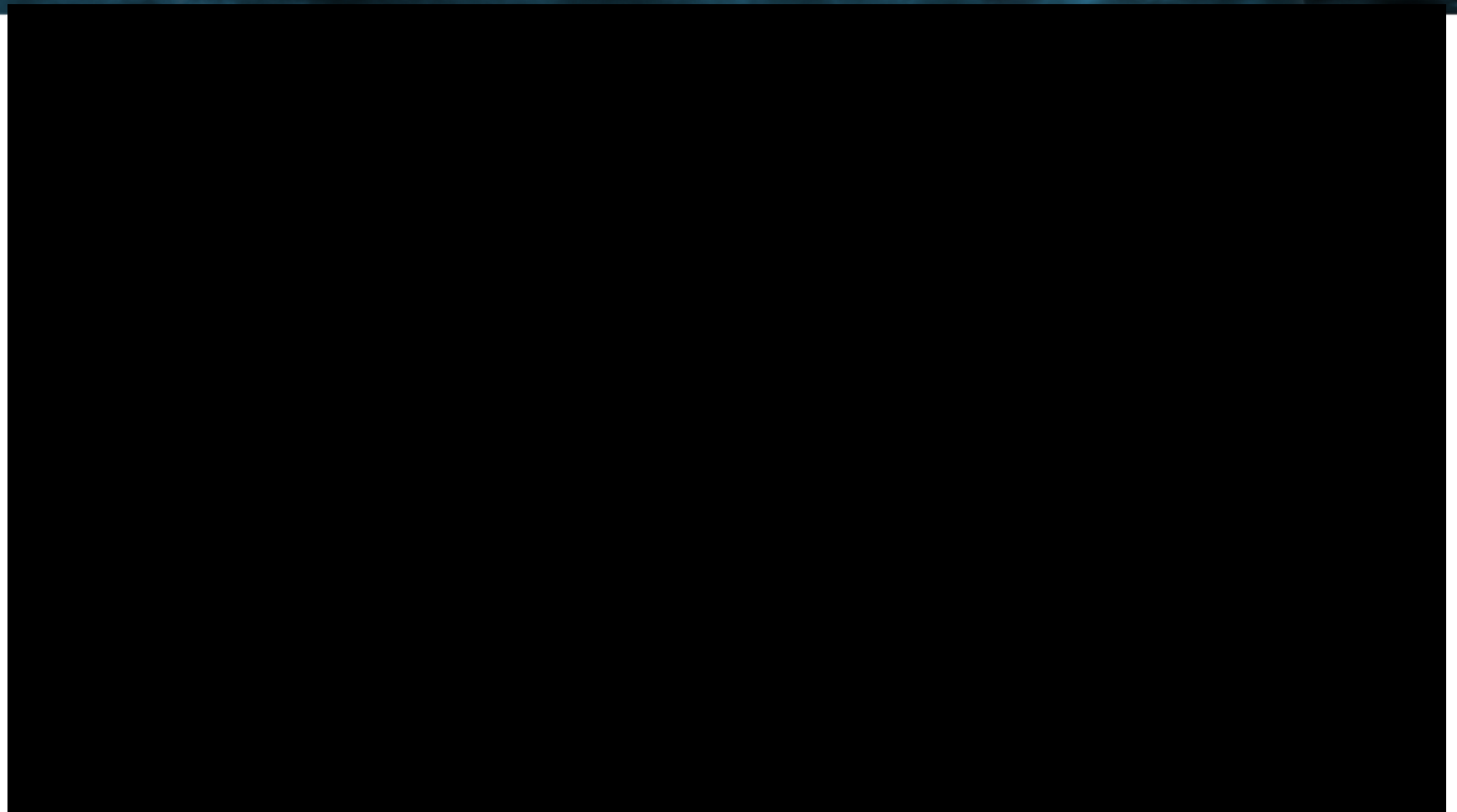


DEMO 2: Malformed Flow Rule Generation



3 The AM instructs the channel agent to disconnect the switch A p action

DEMO 2: Malformed Flow Rule Generation

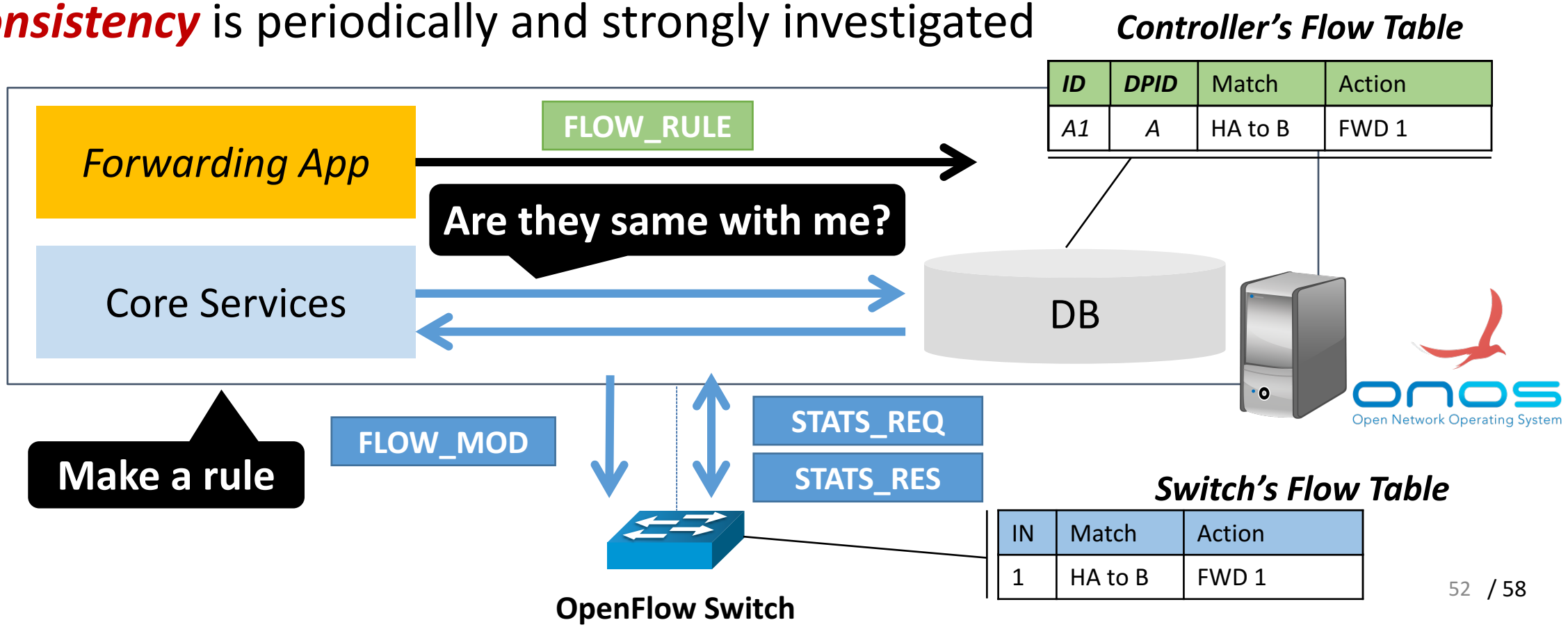


DEMO 2: Malformed Flow Rule Generation

- Feasible to OpenDaylight oxygen (latest version)
- ***Why?***
  - Improper exception handling in the handshake process
  - Absence of malformed flow rule management
- ***How to defend?***
  - Detecting the infinite failures and resolving root causes
  - Filtering an input that has incompatible fields

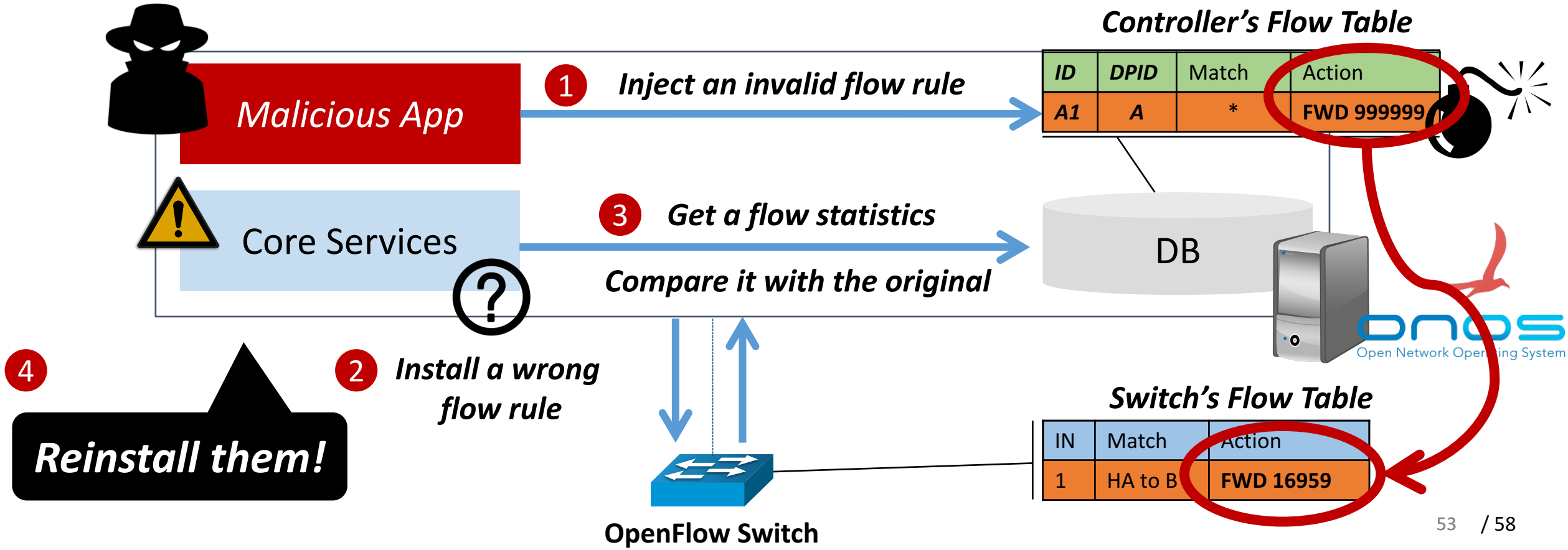
Flow Synchronization in ONOS

- ONOS synchronizes the internal flow tables with switches using ***flow statistics***
- ***Consistency*** is periodically and strongly investigated

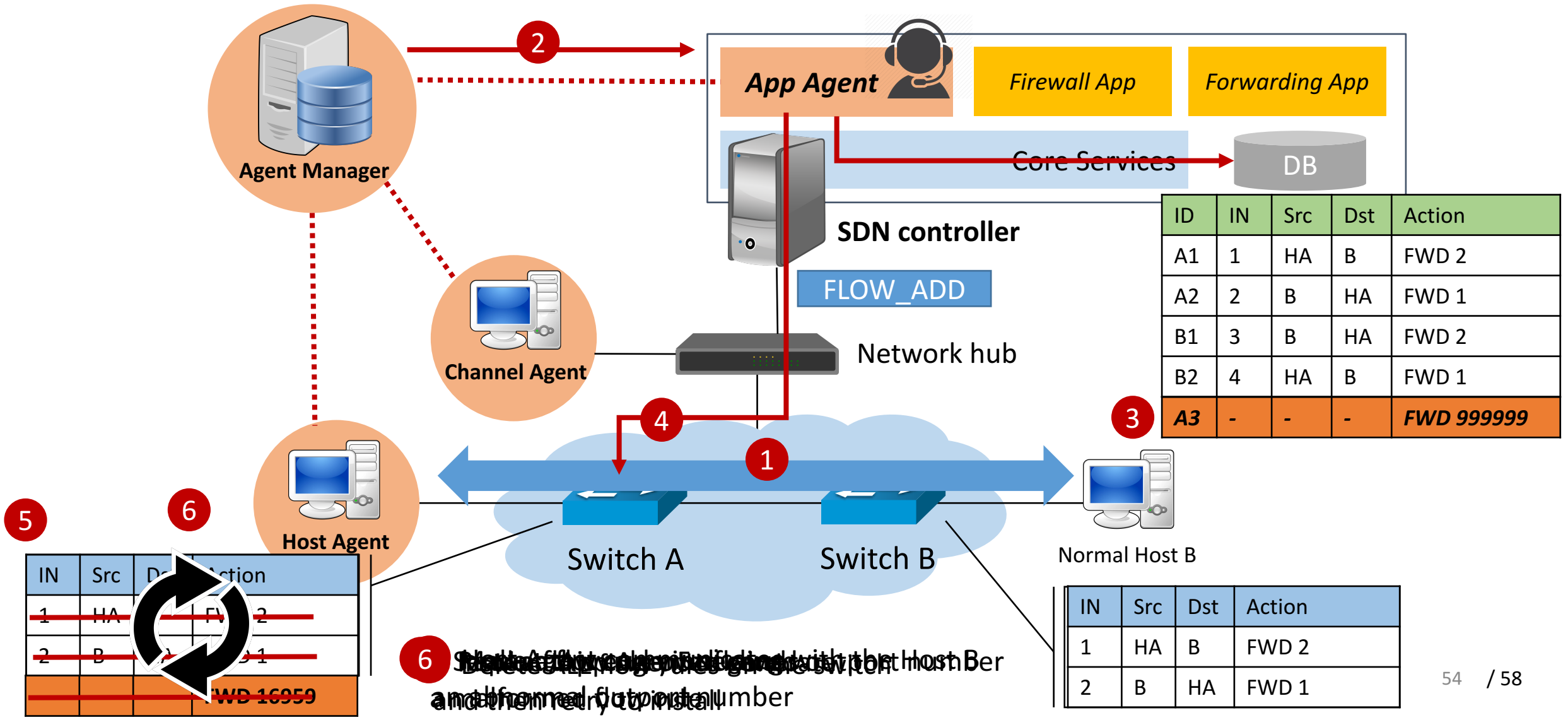


Attack Strategy: Exploit the synchronization!

- If consistency is broken, ONOS ***removes*** and ***reinstalls*** everything
- Let's ***break the consistency*** by installing a malformed flow rule



DEMO 3: Infinite Flow Rule Synchronization

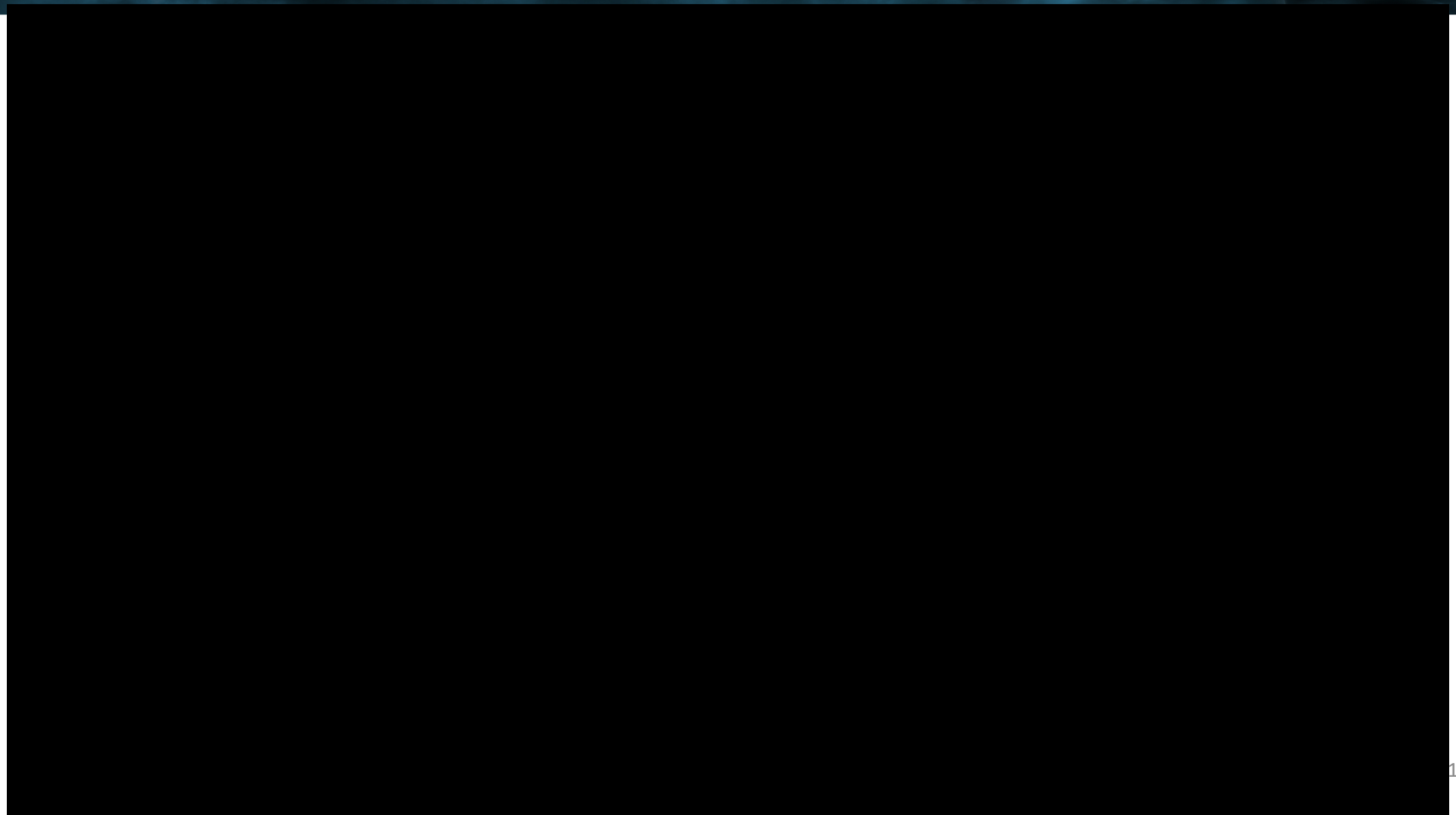


ID	IN	Src	Dst	Action
A1	1	HA	B	FWD 2
A2	2	B	HA	FWD 1
B1	3	B	HA	FWD 2
B2	4	HA	B	FWD 1
A3	-	-	-	***FWD 999999***

IN	Src	Dst	Action
~~1~~	~~HA~~	~~A~~	~~FWD 2~~
~~2~~	~~B~~	~~A~~	~~FWD 1~~
			~~FWD 16959~~

IN	Src	Dst	Action
1	HA	B	FWD 2
2	B	HA	FWD 1

DEMO 3: Infinite Flow Rule Synchronization



DEMO 3: Infinite Flow Rule Synchronization

- Feasible to ONOS 1.13 (latest version)
- ***Why?***
  - **Careless range check** against to field values
  - Meaningless flow synchronization
- ***How to defend?***
  - Thorough range check in critical fields
  - Root cause analysis of synchronization failures

Summary of NEW attack cases

No.	*Attack Name*	*Control Flow Type*	*Controller*
1	**Malformed Flow Rule Generation 1**	**Intra-Controller Flow**	**OpenDaylight**
2	Malformed Flow Rule Generation 2	Intra-Controller Flow	ONOS
3	Flow Rule Inconsistency 1	Asymmetric Flow	ONOS
4	Flow Rule Inconsistency 2	Asymmetric Flow	Floodlight
5	Flow Rule Inconsistency 3	Asymmetric Flow	ONOS
6	**Infinite Flow Rule Synchronization 1**	**Asymmetric Flow**	**ONOS**
7	Infinite Flow Rule Synchronization 2	Asymmetric Flow	ONOS
8	Flow Rule ID Spoofing 1	Asymmetric Flow	Floodlight
9	Flow Rule ID Spoofing 2	Asymmetric Flow	Floodlight

Final Remarks

- Although SDN offers significant benefits as a next-gen networking, a lot of work still needs to be done to improve the security of SDN.
- **DELTA** helps to verify the security of SDN architecture thoroughly.
- **DELTA** fuzzing techniques enable us to discover new vulnerabilities.

DELTA *is now available as an open source project, so anyone can join us! (https://github.com/OpenNetworkingFoundation/delta)*

Q&A

Thanks to **Changhoon Yoon** and **Haney Kang** for helping us make the slides ☺

Acknowledgement

- This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00254, SDN security technology development)
- And also, this work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. B0190-16-2012, Global SDN/NFV OpenSource Software Core Module/Function Development)