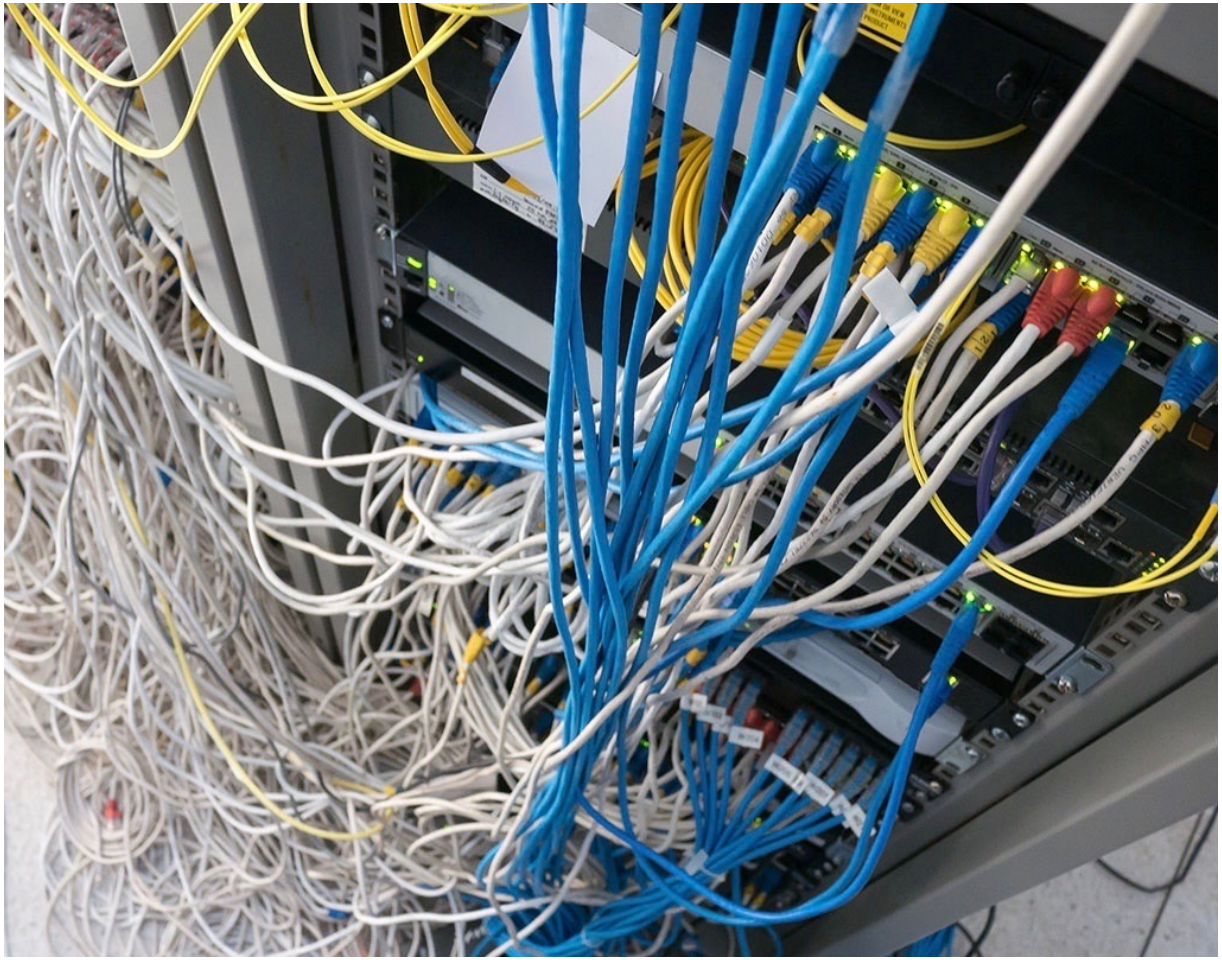
Finding Semantic Bugs in Intent-Based Networking with Fuzzing

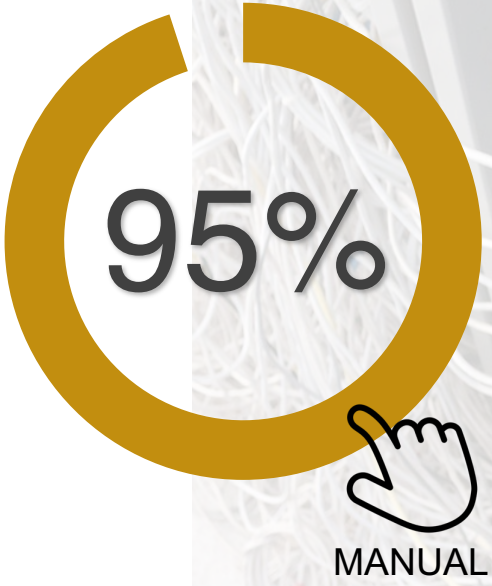
Jiwon Kim

Purdue University

Nov 16, 2023

What is Intent-Based Networking?





wttw NEWS

BUSINESS

Facebook Blames Outage on Error During Routine Maintenance

Associated Press | October 6, 2021 11:05 am

BLEEPINGCOMPUTER

NEWS | DOWNLOADS | VPNS | VIRUS REMOVAL GUIDES | TUTORIALS | DEALS | FORUMS | MORE

South Korean telco KT suffers nationwide outage after routing error

October 25, 2021 12:20 PM 0

BBC NEWS

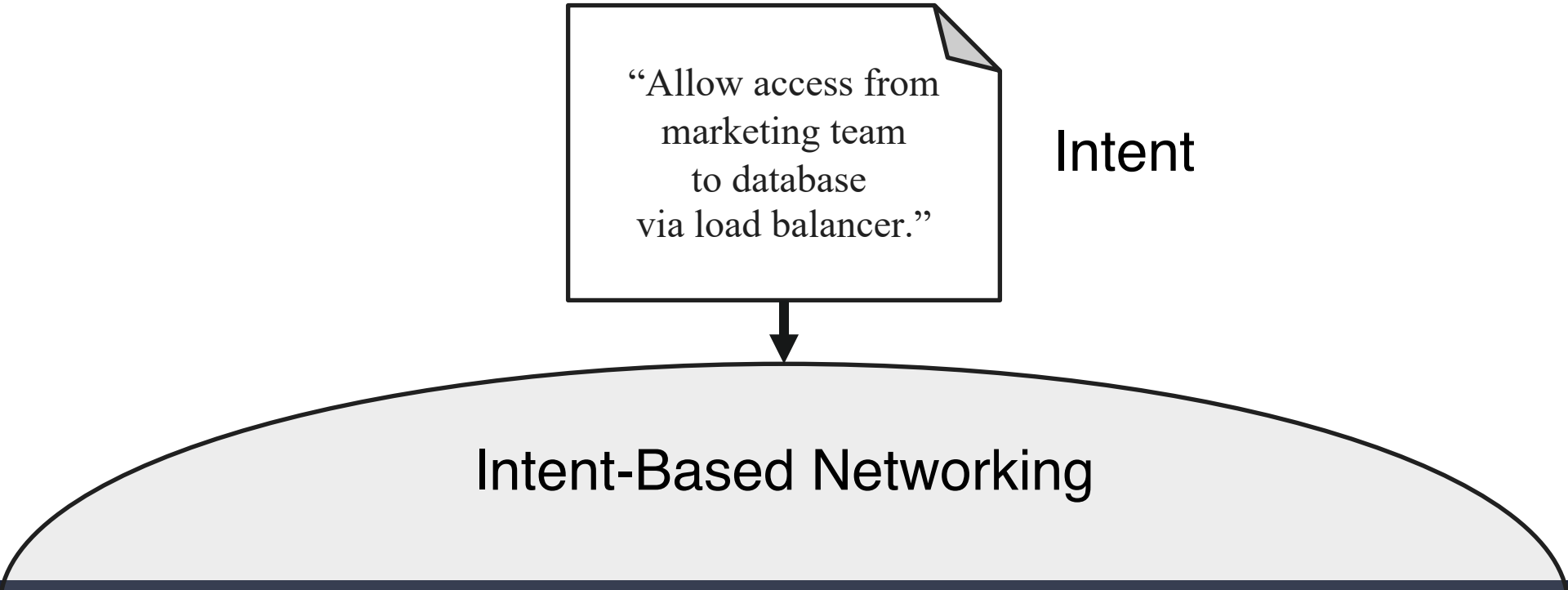
Home | War in Ukraine | Climate | Video | World | US & Canada | UK | Business | Tech | Science | More

Europe | Guernsey

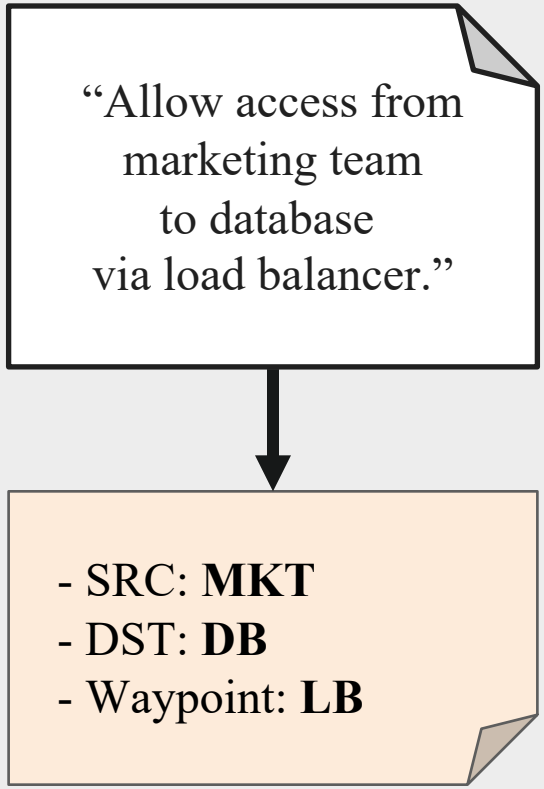
Human error behind network outage, Sure confirms

3 February

[1] Cisco, https://www.cisco.com/c/en/us/solutions/automation/network-automation.html


“Allow access from
marketing team
to database
via load balancer.”
Intent
Intent-Based Networking

Translation



Compilation

Network-level Intent(s)

- SRC: **MKT**
- DST: **DB**
- Waypoint: **LB**

IN	SRC	DST	OUT
1	MKT1	DB	LB
2	MKT2	DB	LB

Network Object(s)

Activation

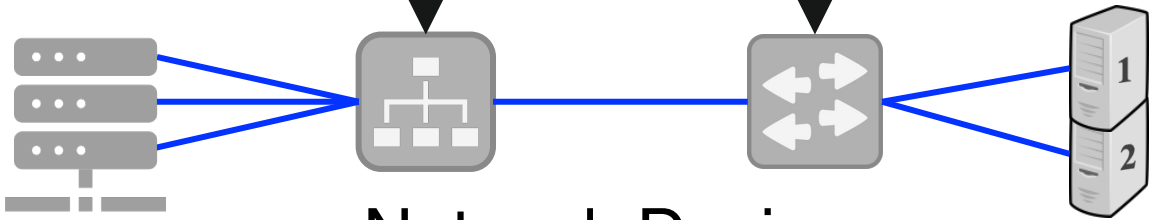
```
config {
  service: db.aa.com
  vip: DB
  port: 3306
  backends: [DB1, 2, 3]
}
```

IN	SRC	DST	OUT
1	MKT1	DB	LB
2	MKT2	DB	LB

Network Object(s)

Control Plane

Data Plane



Network Devices

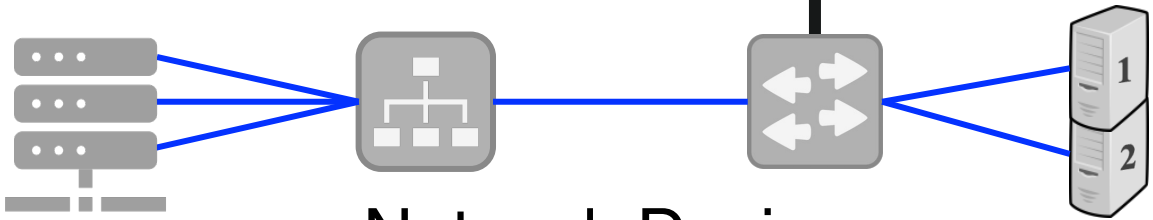
Monitoring

Telemetry Data

```
[S1] RX/TX
MKT1:   20mbps/20kbps
MKT2:  980mbps/20kbps
LB:     40kbps/ 1gbps
```

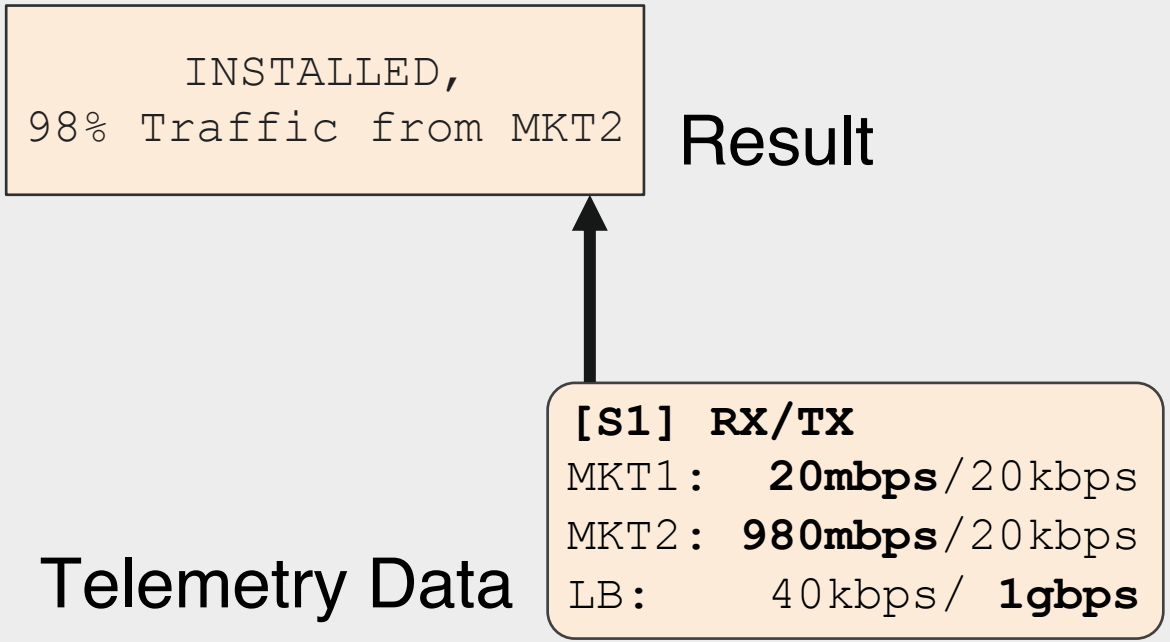
Control Plane

Data Plane

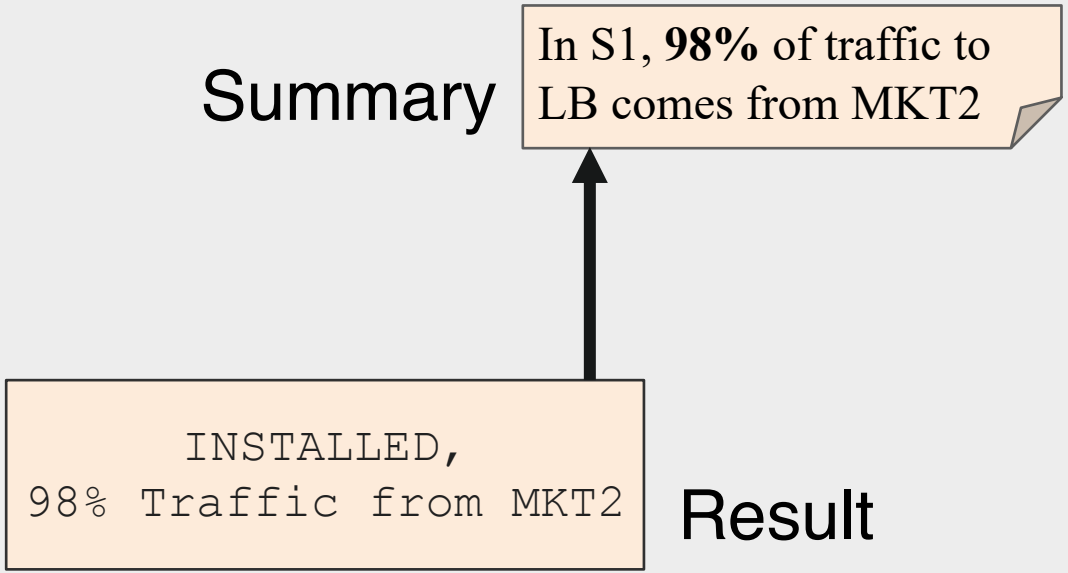


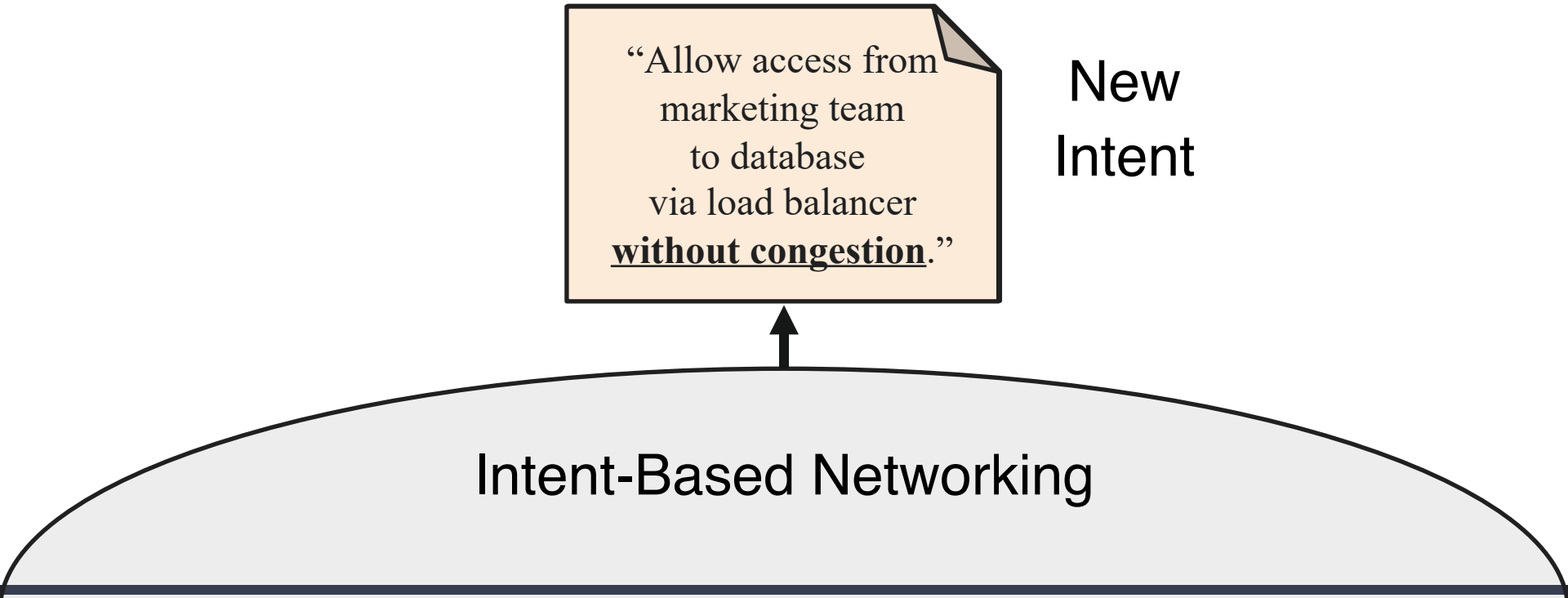
Network Devices

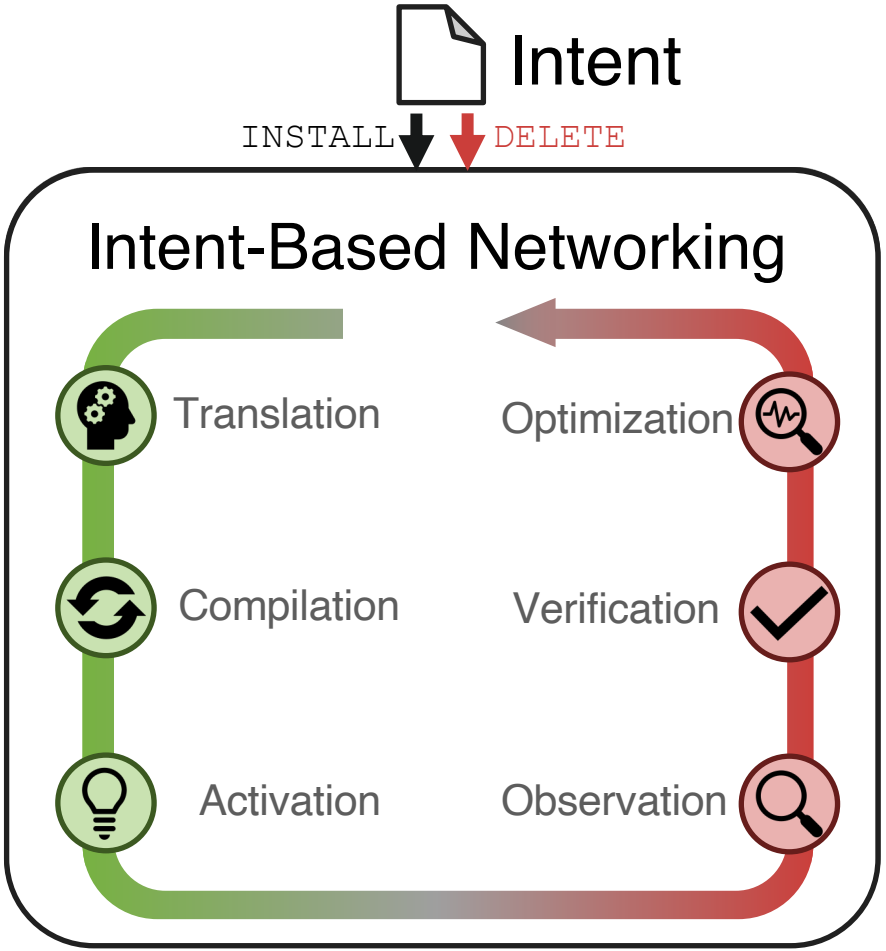
Verification



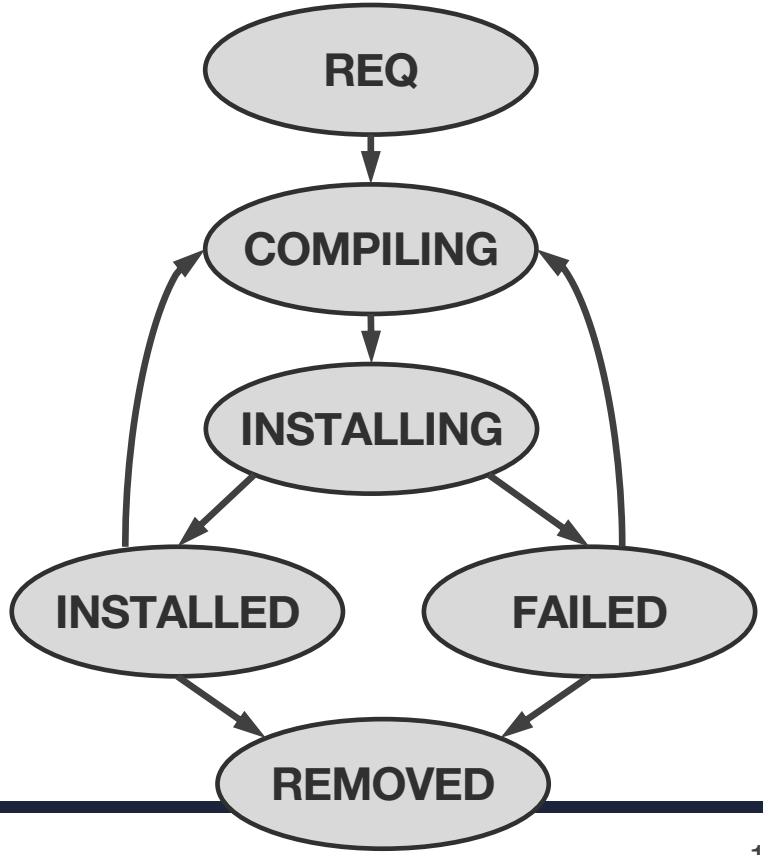
Optimization



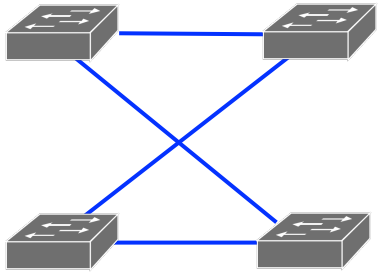

"Allow access from
marketing team
to database
via load balancer
without congestion."
New
Intent
Intent-Based Networking



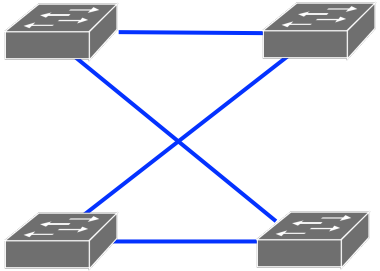
Intent State Machine



SDN and IBN

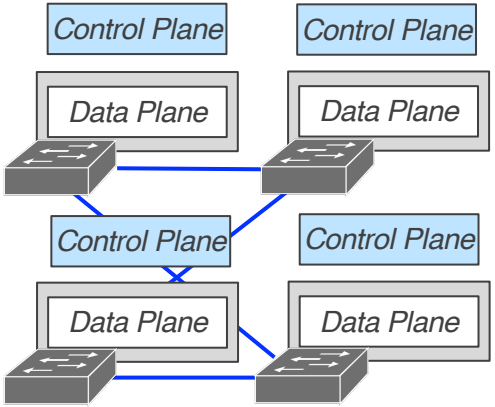


Software-Defined Networking

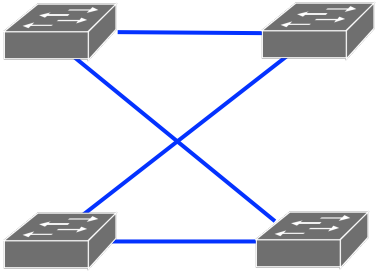


Intent-Based Networking

SDN and IBN

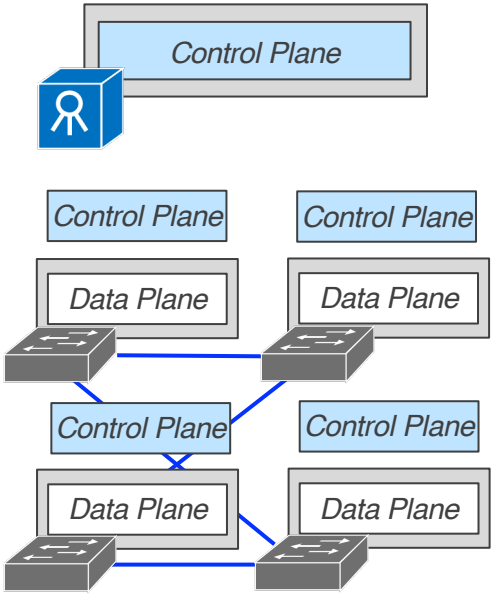


Software-Defined Networking

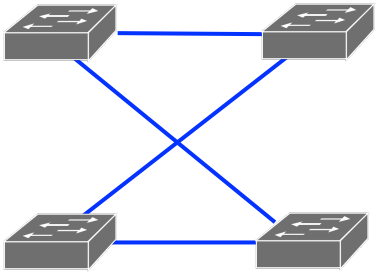


Intent-Based Networking

SDN and IBN

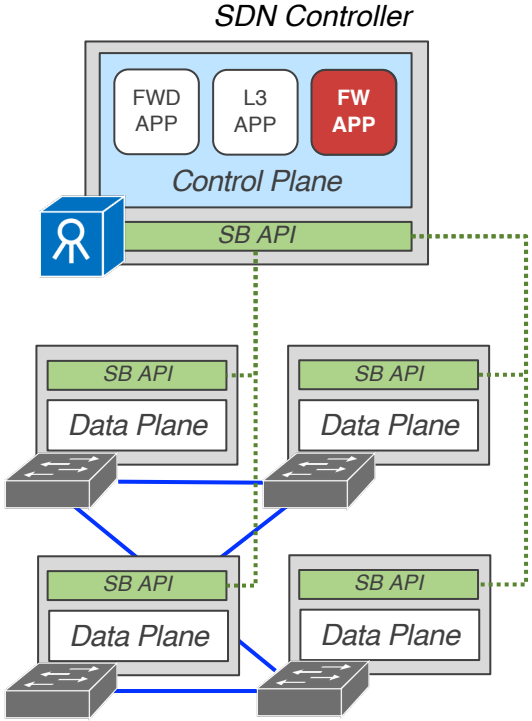


Software-Defined Networking

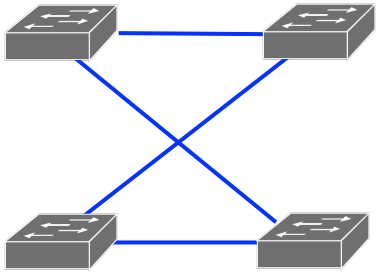


Intent-Based Networking

SDN and IBN

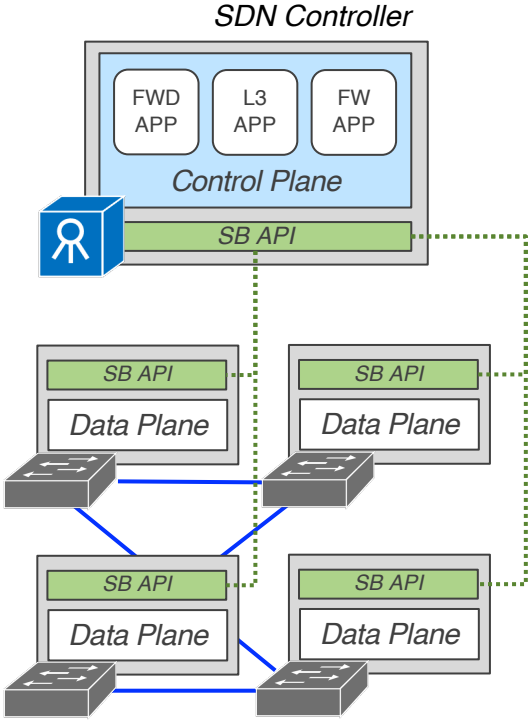


Software-Defined Networking

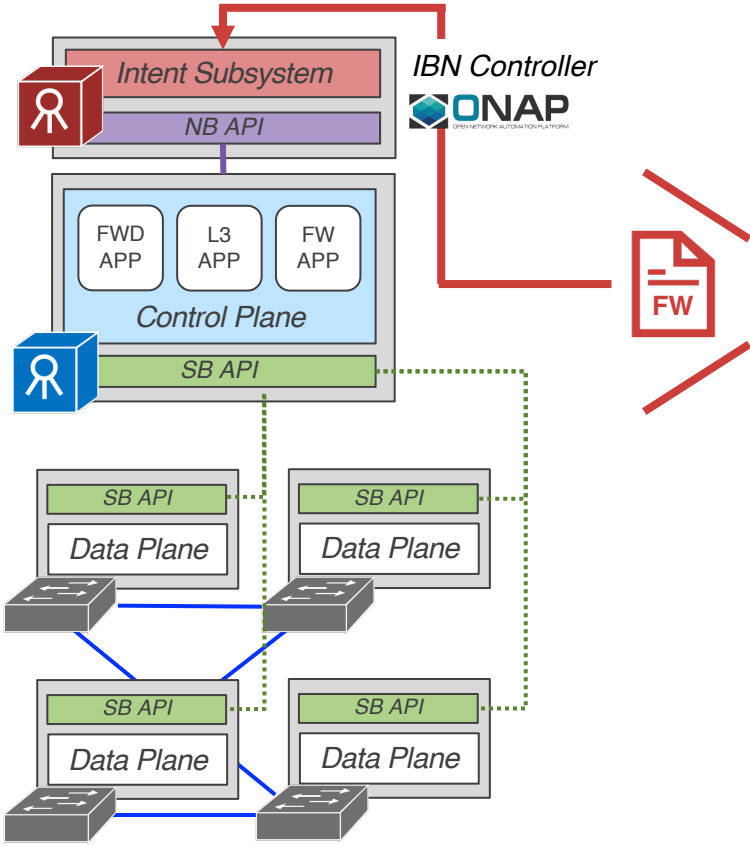


Intent-Based Networking

SDN and IBN

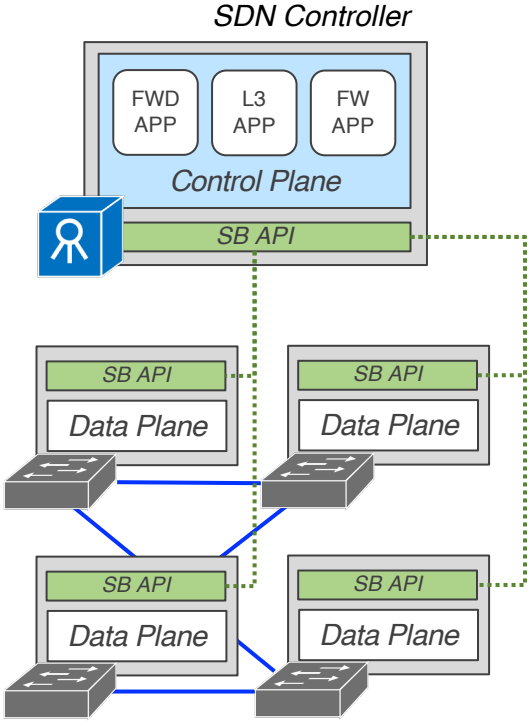


Software-Defined Networking

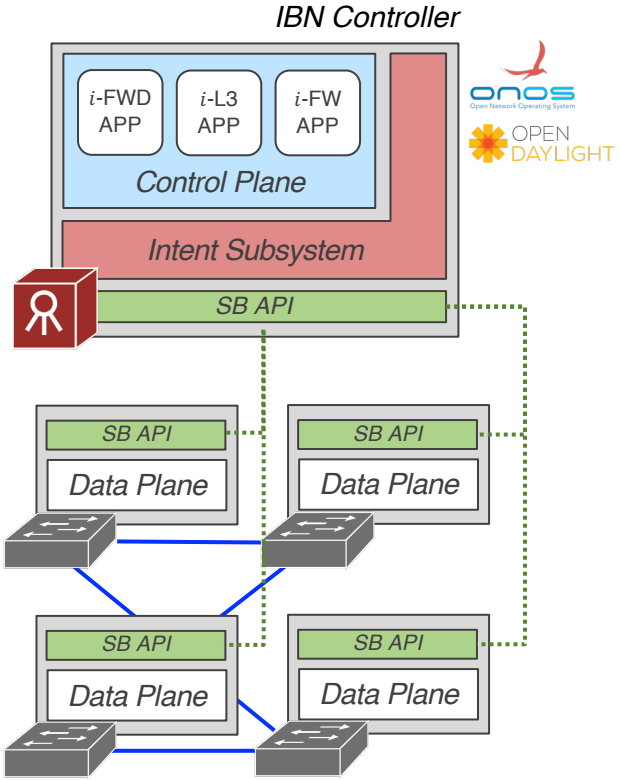


Intent-Based Networking

SDN and IBN

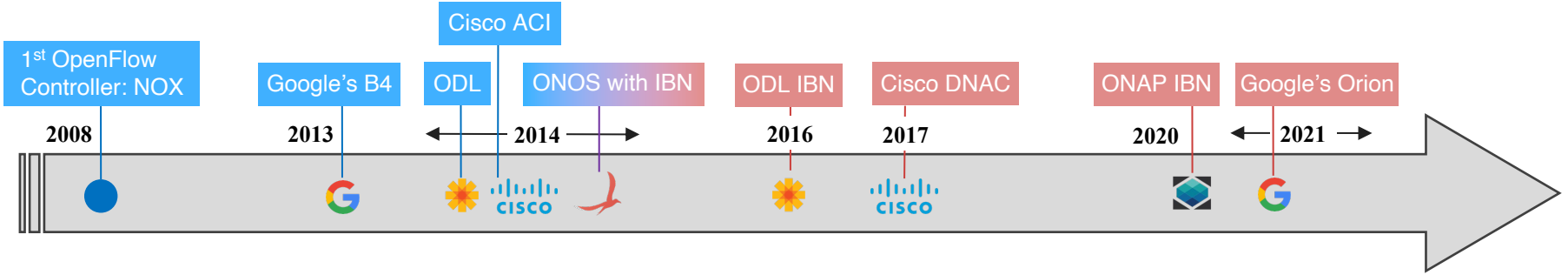


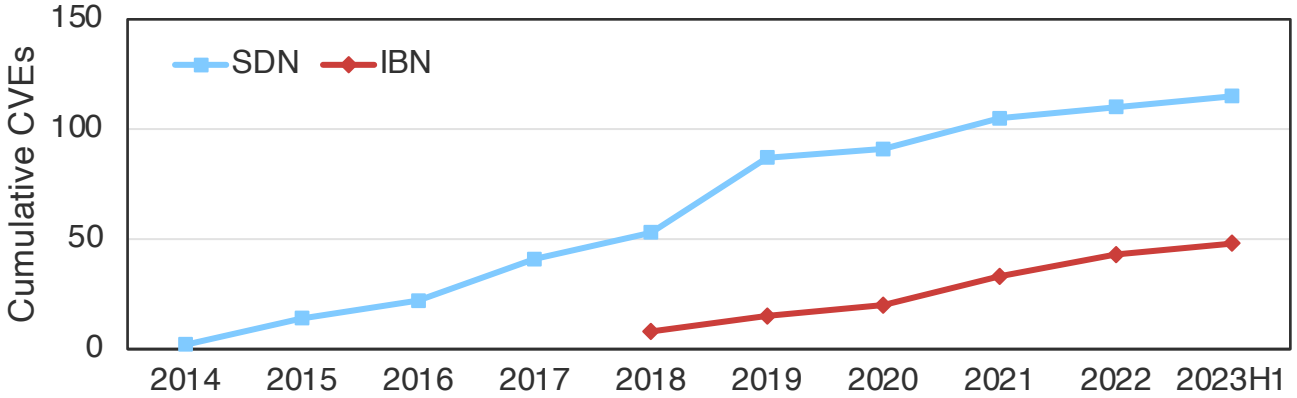
Software-Defined Networking



Intent-Based Networking

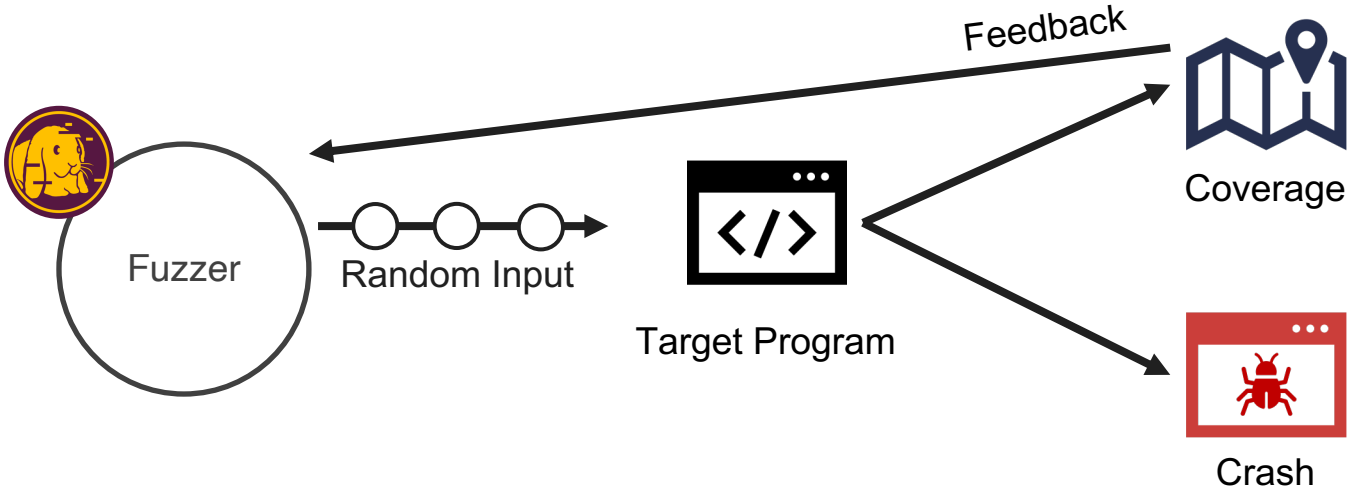
Vulnerabilities in SDN and IBN



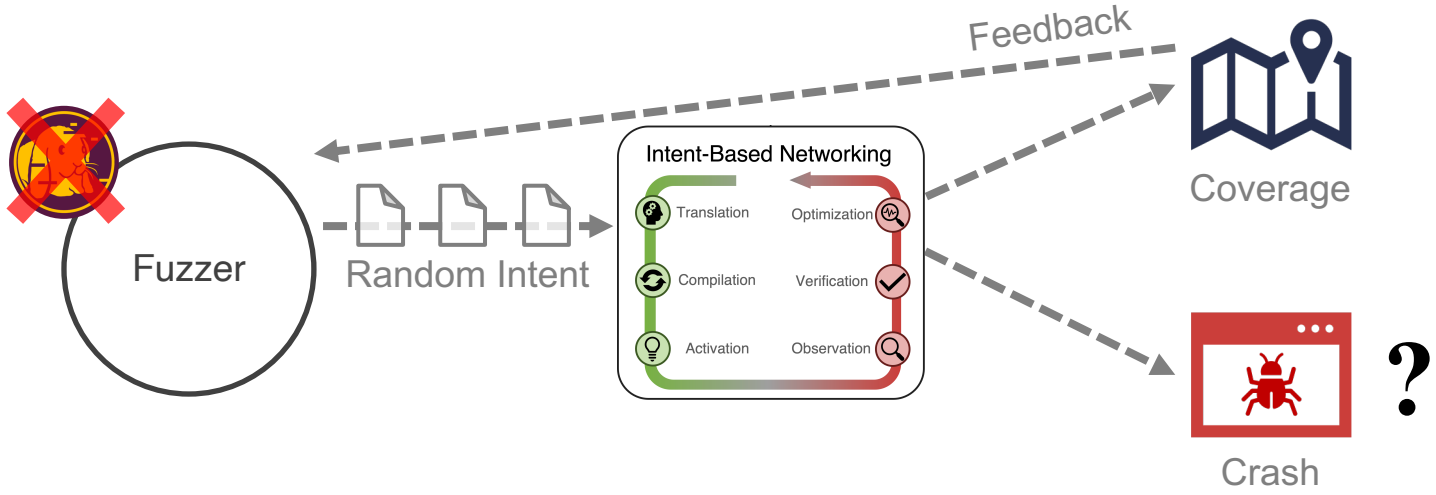


Can we run a *fuzzer* on Intent-Based Networking?

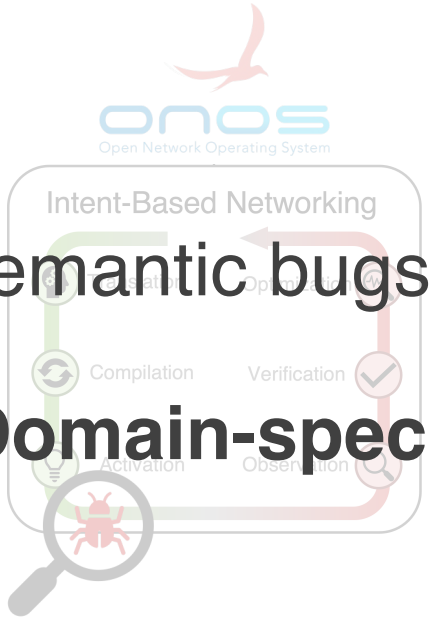
Fuzzing Programs

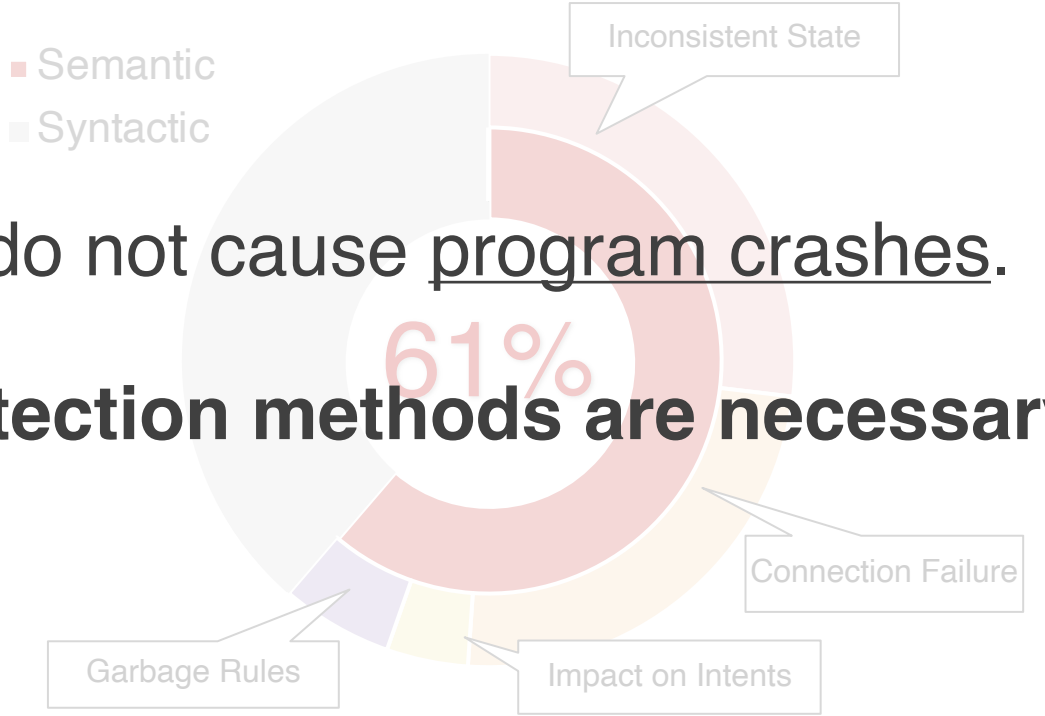


Fuzzing IBN [1/3]



I. Bug Study in ONOS IBN

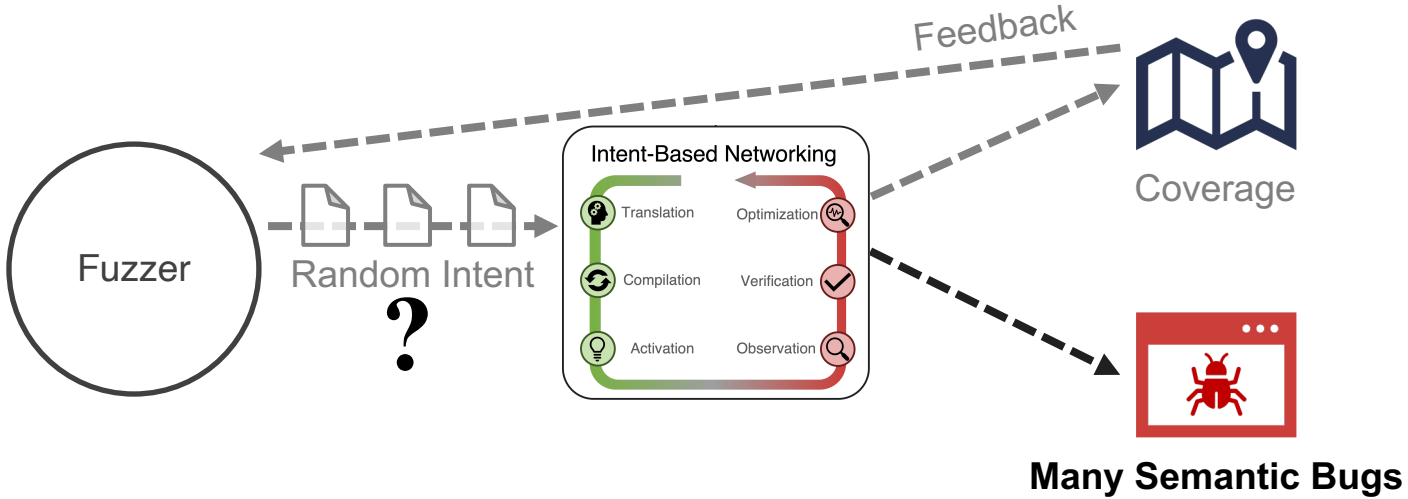




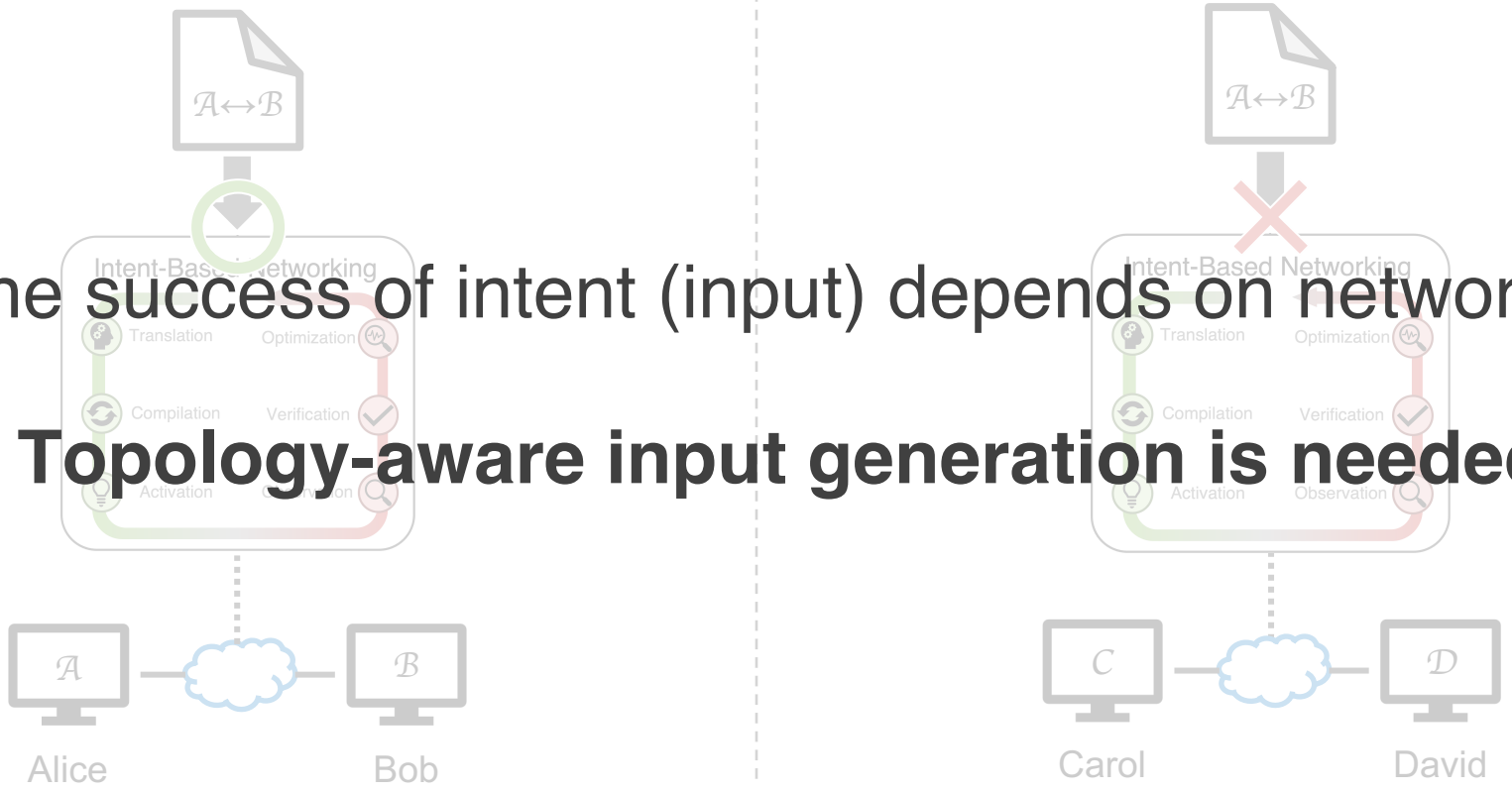
Semantic bugs often do not cause program crashes.

→ **Domain-specific detection methods are necessary.**

Fuzzing IBN [2/3]



II-1. Limitation in Input Generation



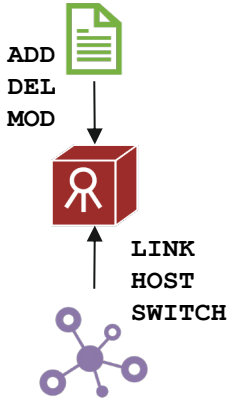
The success of intent (input) depends on network.

→ **Topology-aware input generation is needed.**

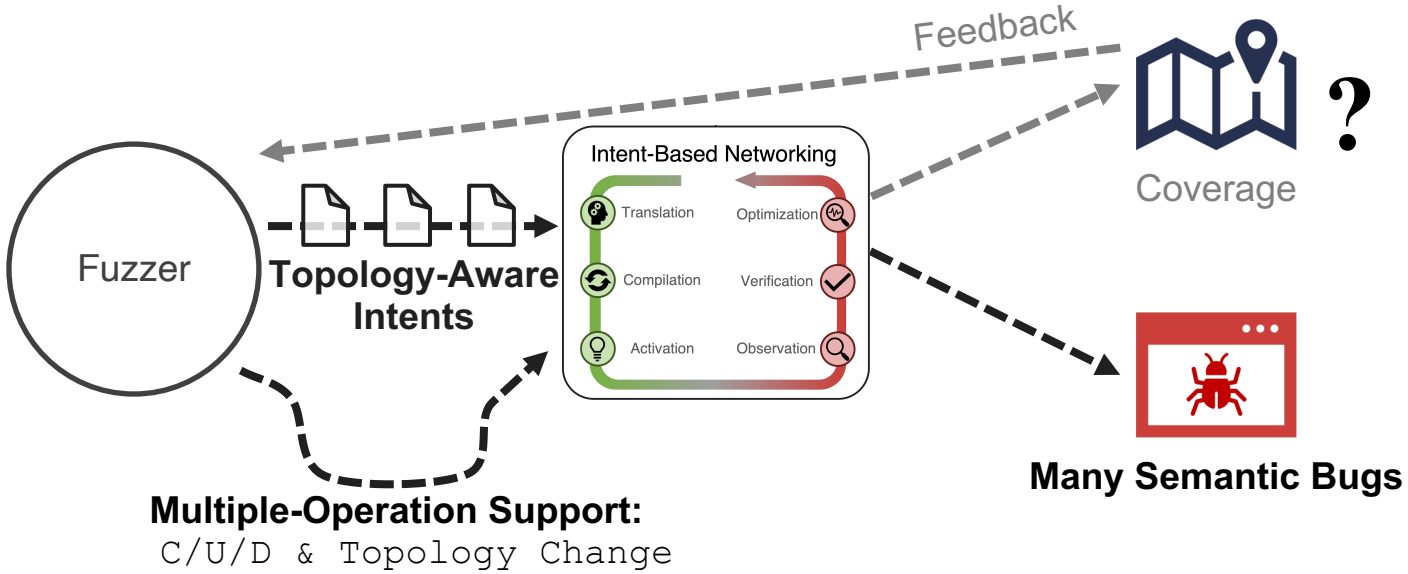
II-2. Bug Study in ONOS IBN

Table 1: Bug analysis of the ONOS intent subsystem.

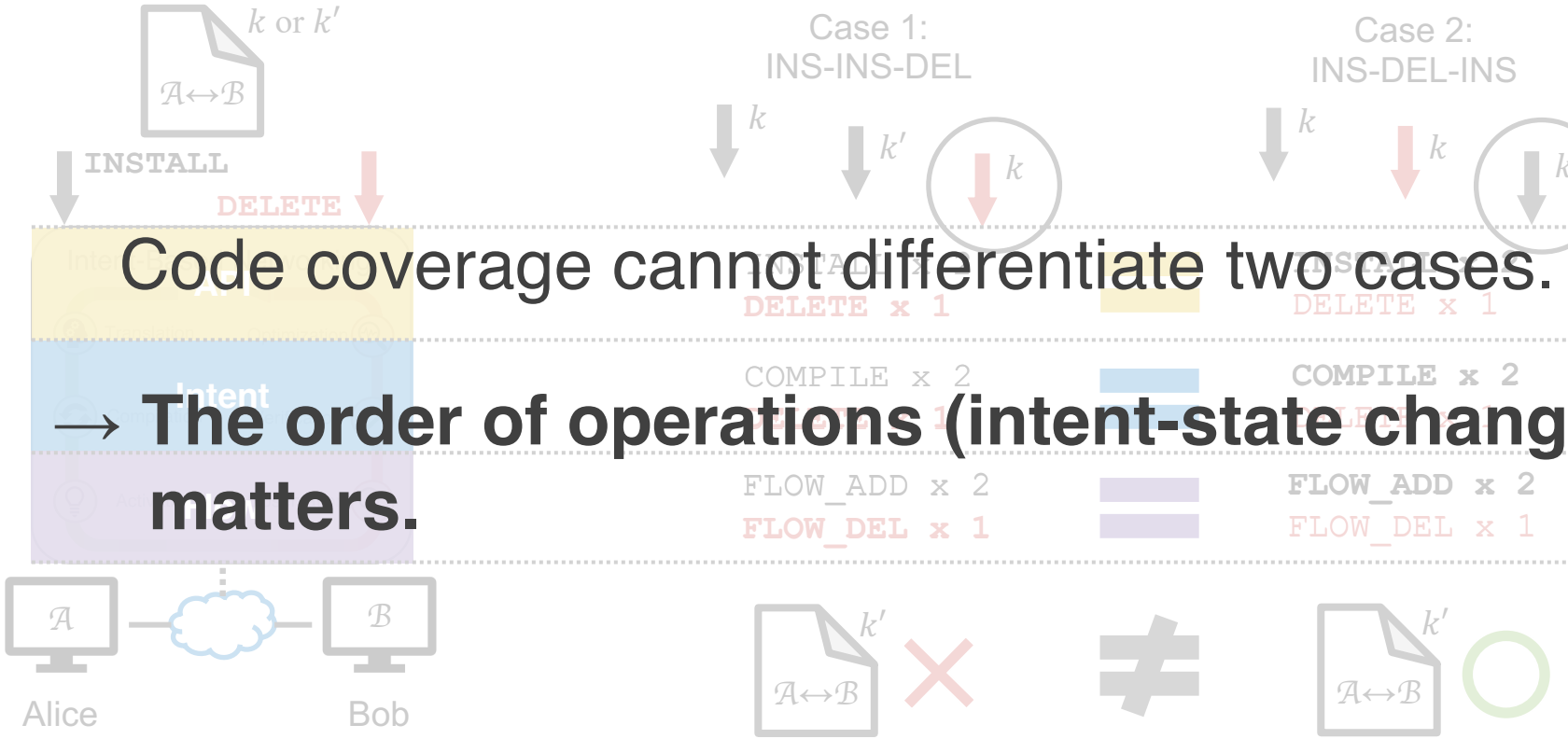
Type	Code	Impact	Detection Mechanism	# Bugs Based on Intent Operation of Root Cause				
				submit	withdraw	purge	topo-change	Total
Intent syntactic bug	**SYN1**	Denied intent request	Input validation	2	-	-	-	2
	SYN2	Not found	Input validation	9	-	-	-	9
	SYN3	Wrong intent data	Input validation	3	-	-	-	3
	SYN4	Corrupt intent	Input validation	1	-	-	-	1
Other syntactic bug	**SYN5**	Controller shutdown	Application agent	1	-	-	-	1
	SYN6	Resource exhaustion	Resource agent	1	2	-	-	3
	SYN7	Topology disband	Application agent	2	1	-	-	3
	SYN8	Throughput drop	Performance test	15	-	-	1	16
	SYN9	Exceptions	Log detection	29	6	2	5	42
Intent semantic bug	**SEM1**	Inconsistent intent state	Control-Plane (CP) test	17	10	7	16	50
	SEM2	Failure in connectivity	Data-Plane (DP) test	29	2	-	14	45
	SEM3	Impact on existing intent	CP/DP tests	6	2	-	-	8
	SEM4	Garbage flow rules	Flow-Intent mapping	6	4	-	1	11
Total				121	27	9	37	194 (186[2])



Fuzzing IBN [3/3]



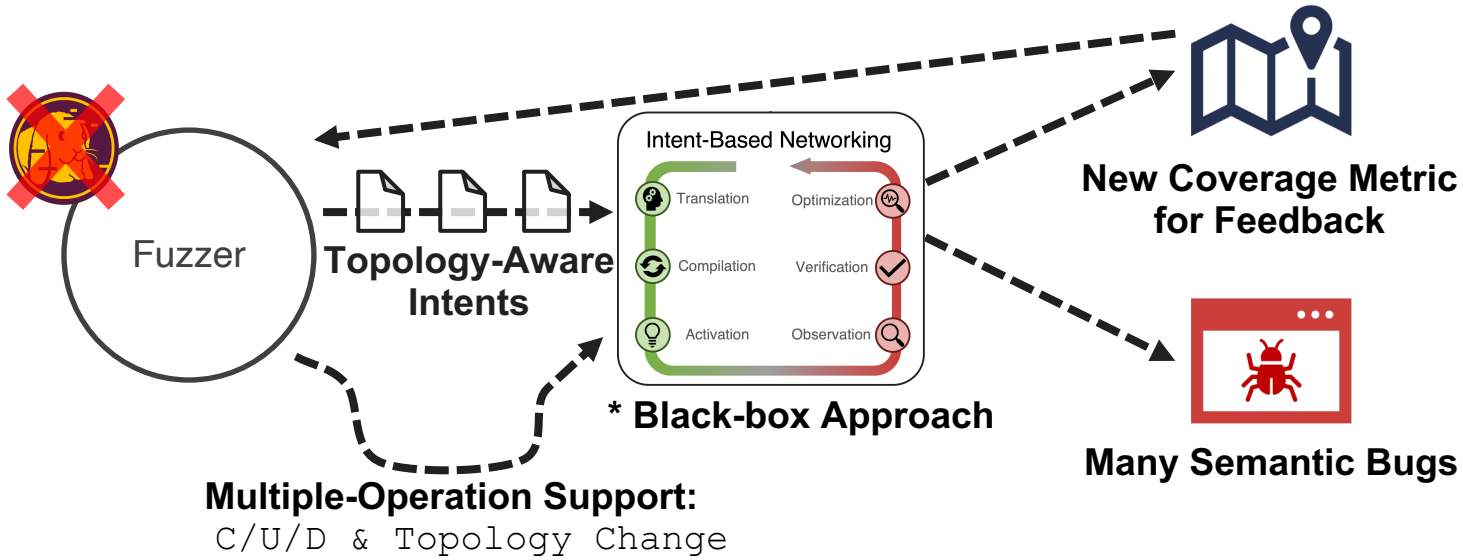
III. Limitation in Code-Coverage Guidance



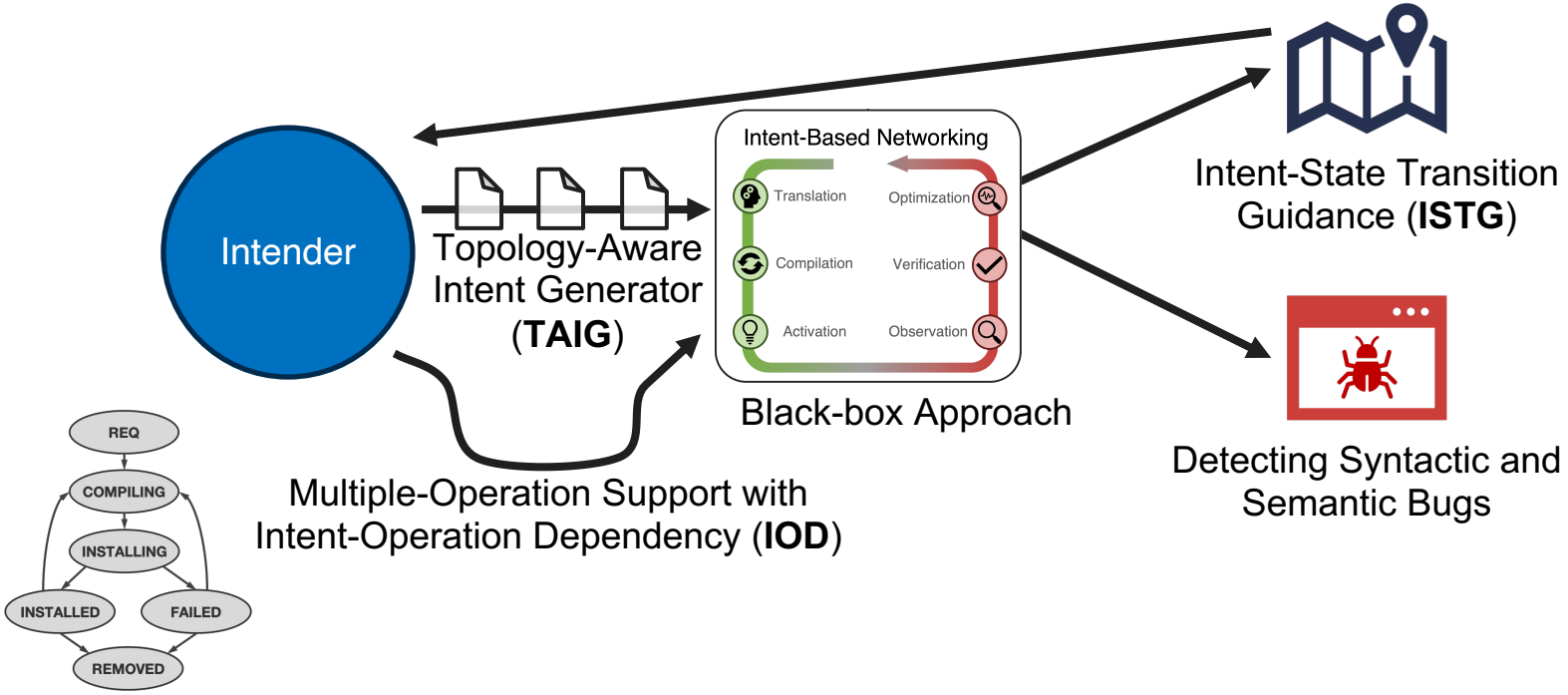
Code coverage cannot differentiate two cases.

→ **The order of operations (intent-state changes) matters.**

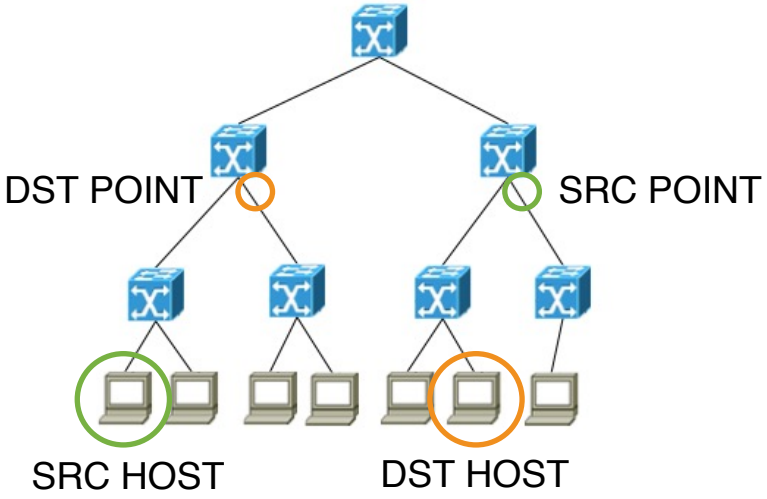
Limitations in Fuzzing IBN



Intender: Fuzzing IBN



Topology-Aware Intent Generator [1/3]



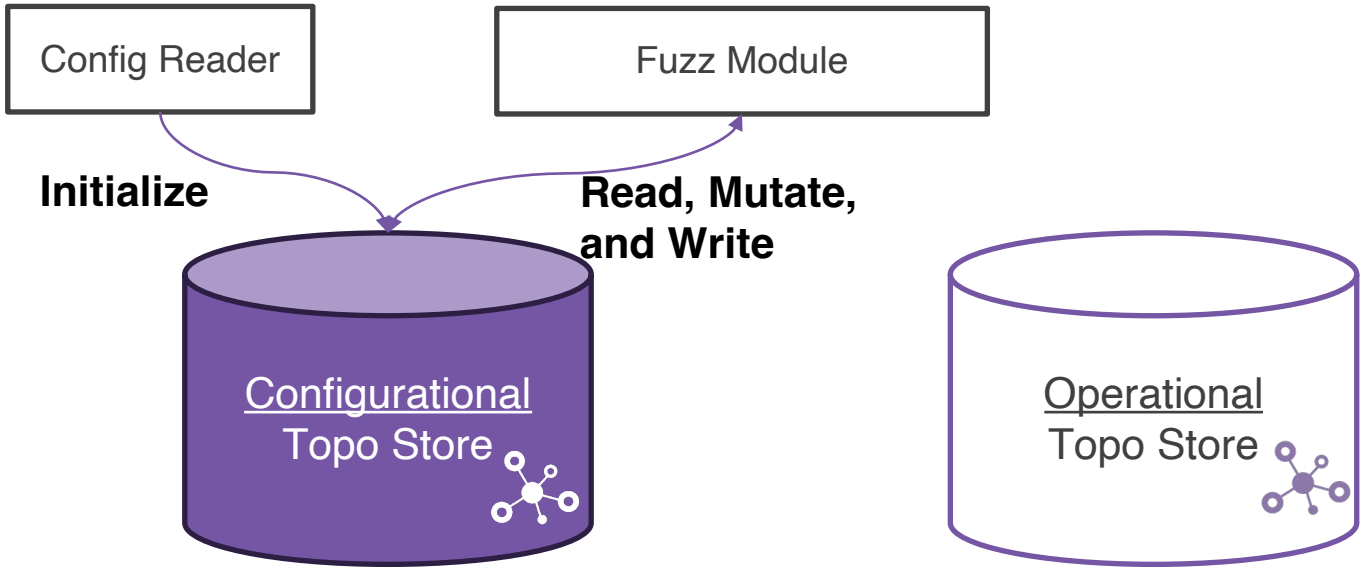
PointToPoint Intent

```
"intent": {
    "type": "PointToPointIntent",
    "appId": "org.onosproject.null",
    "priority": 55,
    "ingressPoint": {"device": "of:0000000000000202", "port": "2"}},
    "egressPoint":  {"device": "of:0000000000000201", "port": "2"}}
}
```

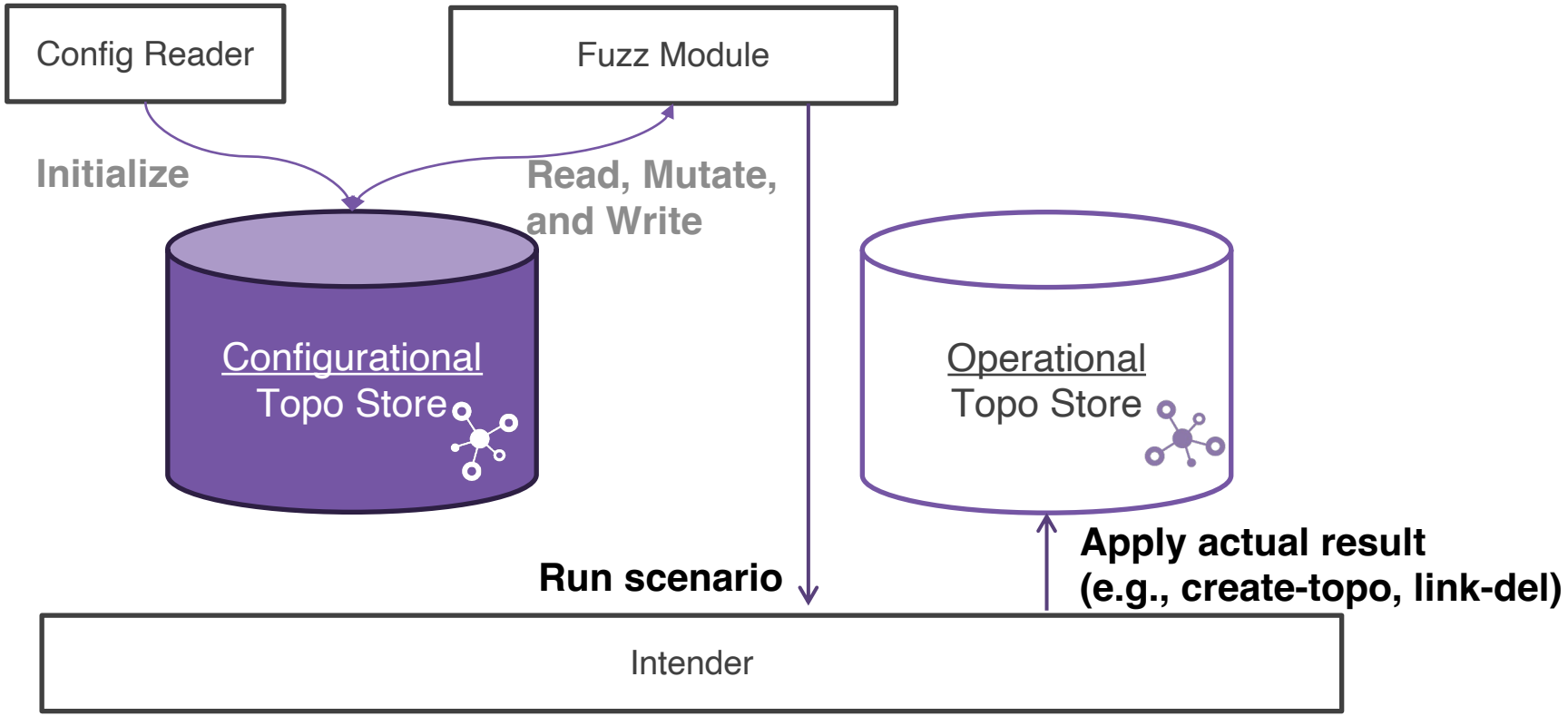
HostToHost Intent

```
"intent": {
    "type": "HostToHostIntent",
    "appId": "org.onosproject.null",
    "priority": 55,
    "one": "12:FB:8D:0B:04:42/None",
    "two": "92:BF:DE:1D:F3:06/None"
}
```

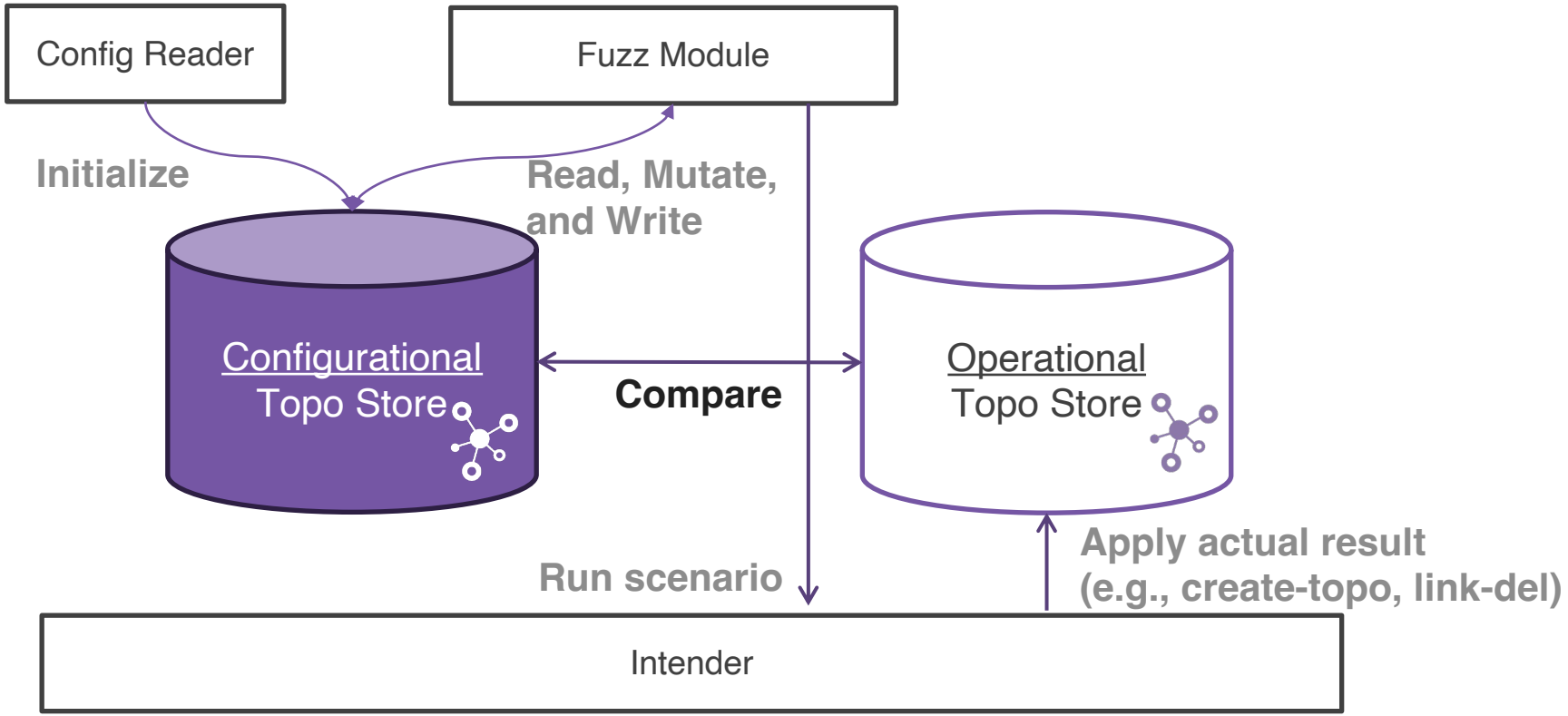
Topology-Aware Intent Generator (cont'd) [1/3]



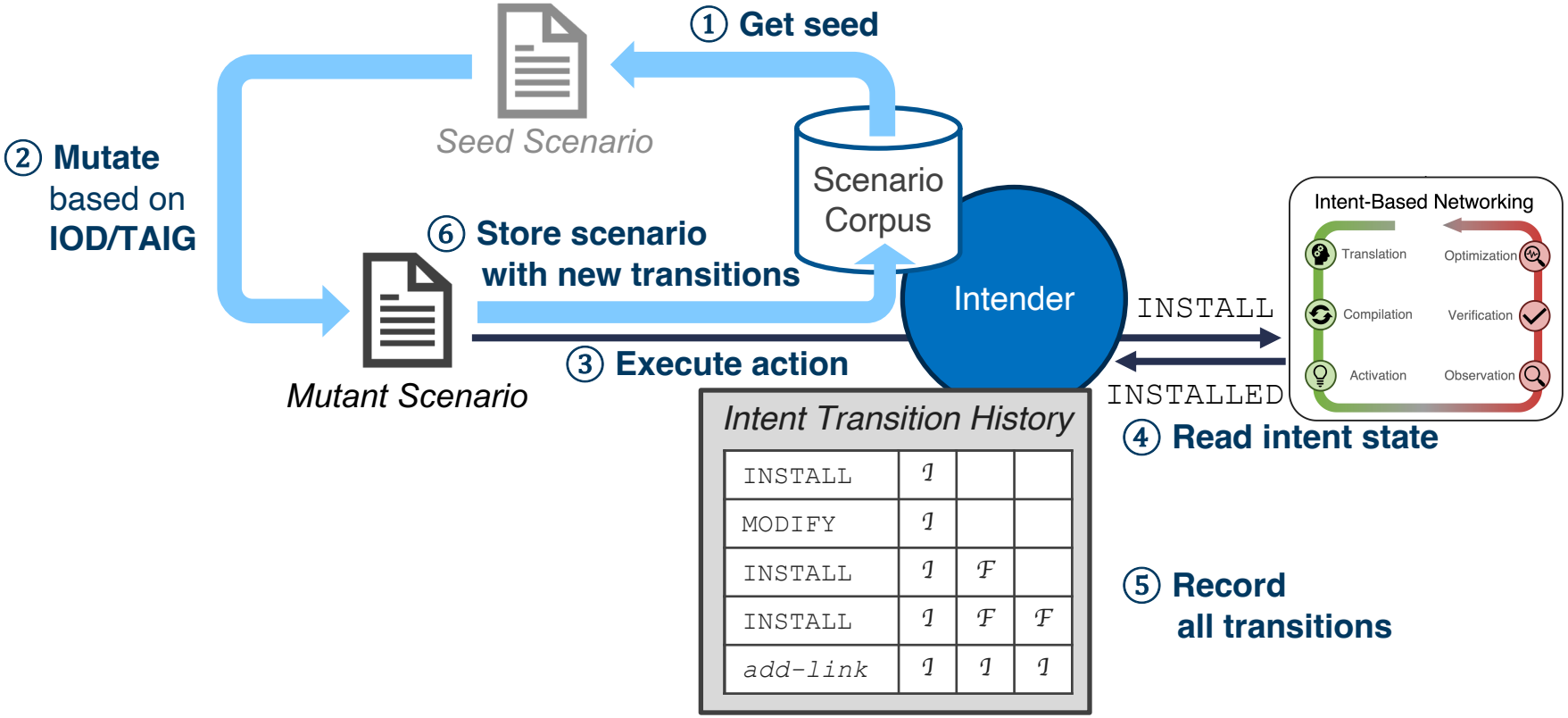
Topology-Aware Intent Generator (cont'd) [1/3]



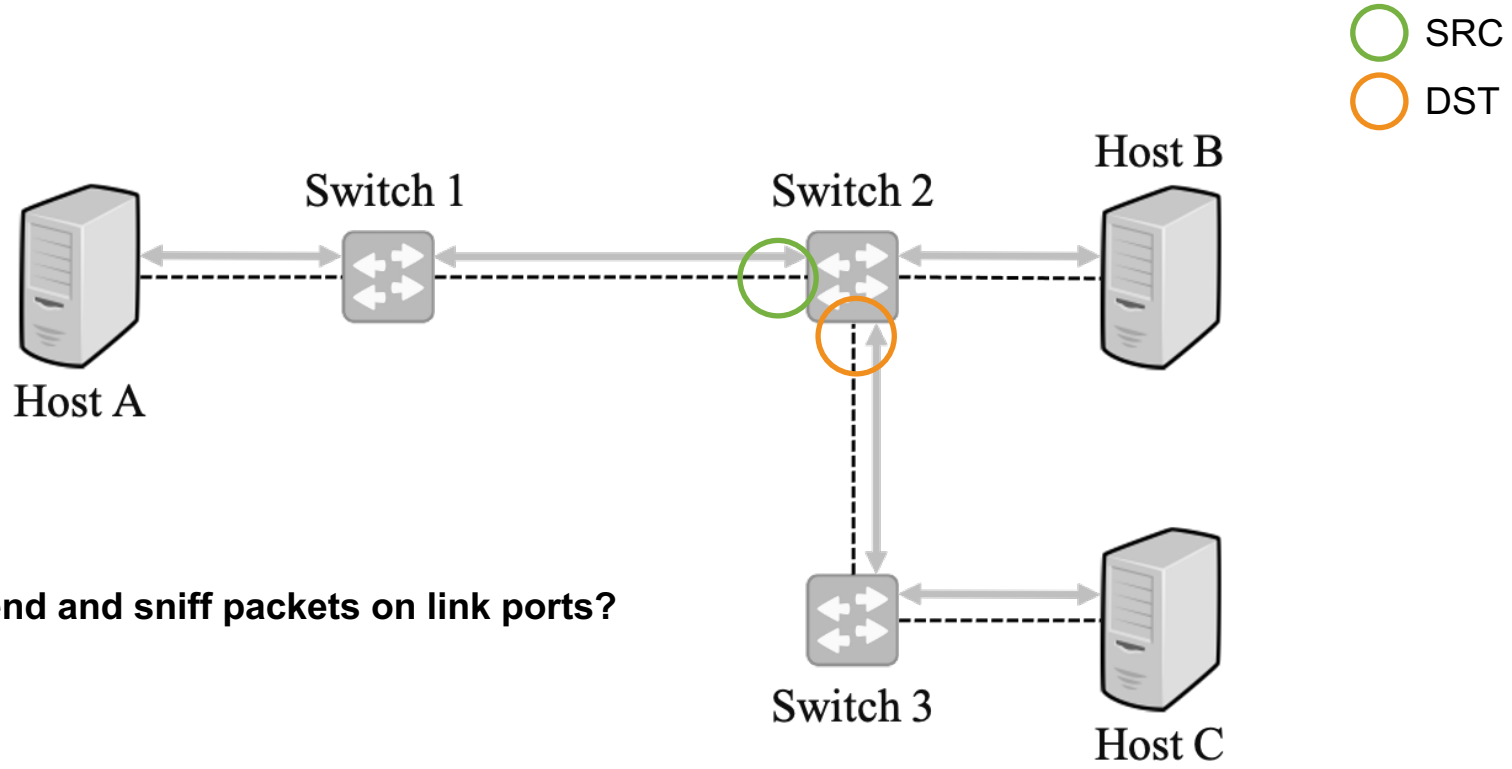
Topology-Aware Intent Generator (cont'd) [1/3]



Intent-State Transition Guidance (ISTG) [2/3]

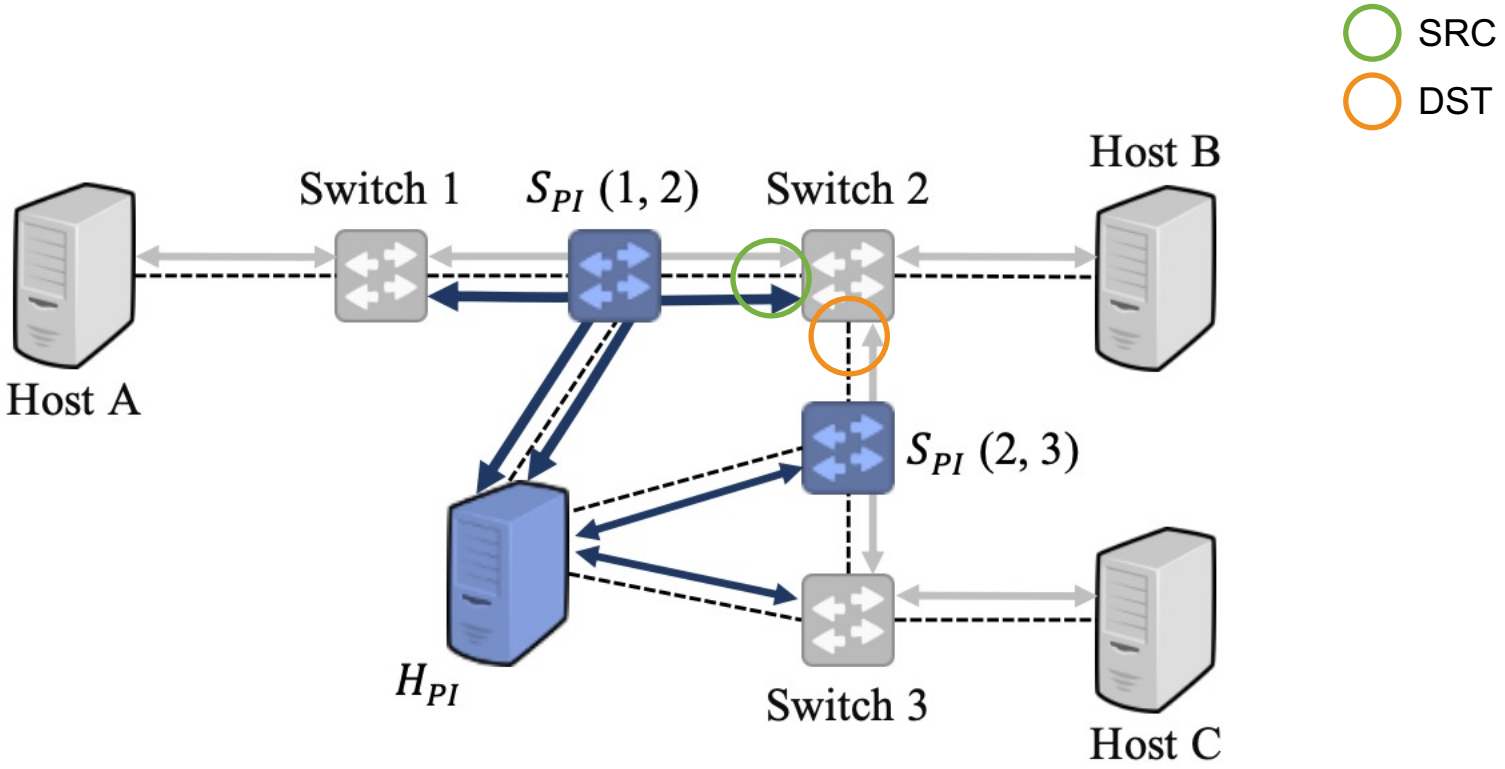


Detecting Connection Failure [3/3]

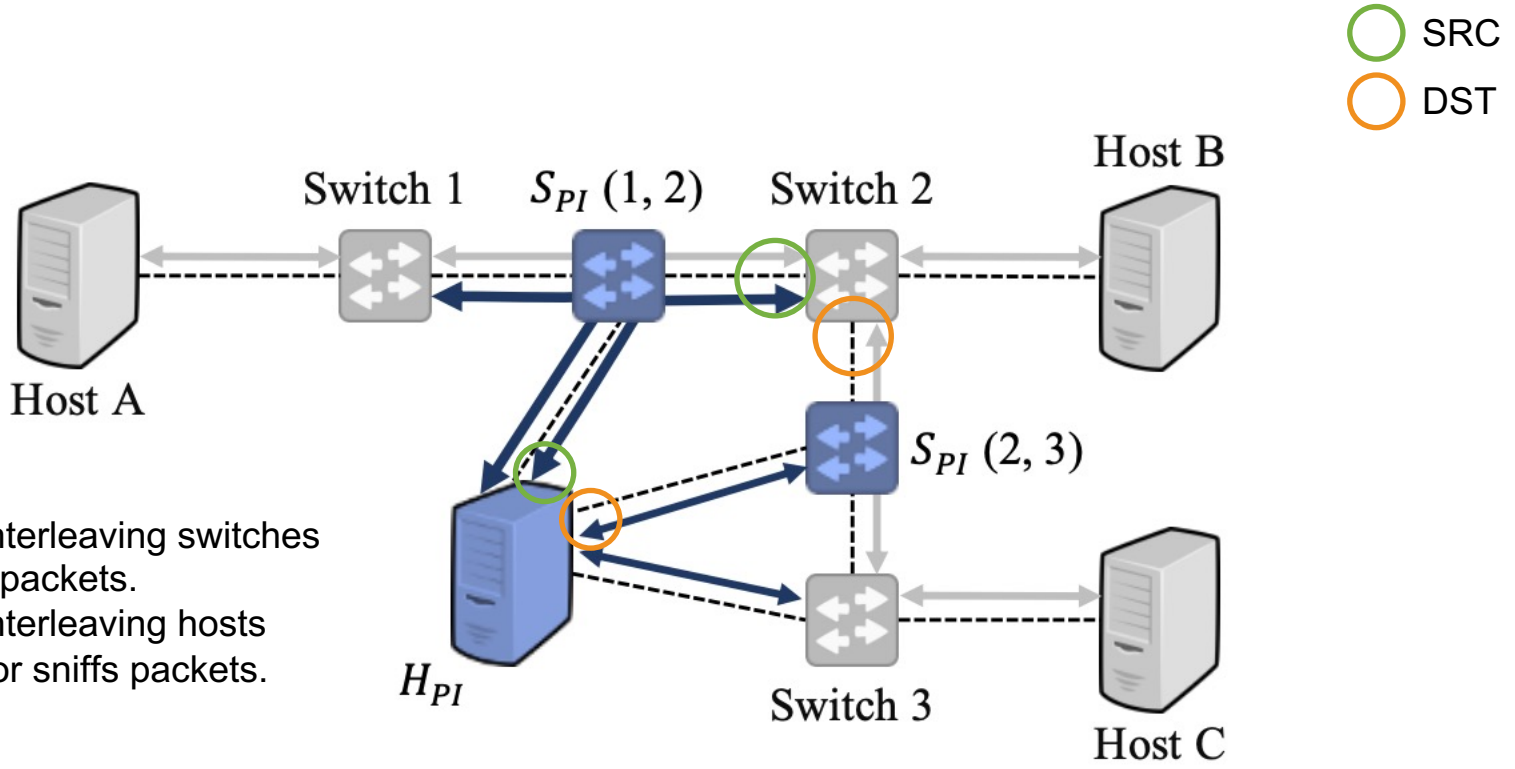


How to send and sniff packets on link ports?

Detecting Connection Failure [3/3]

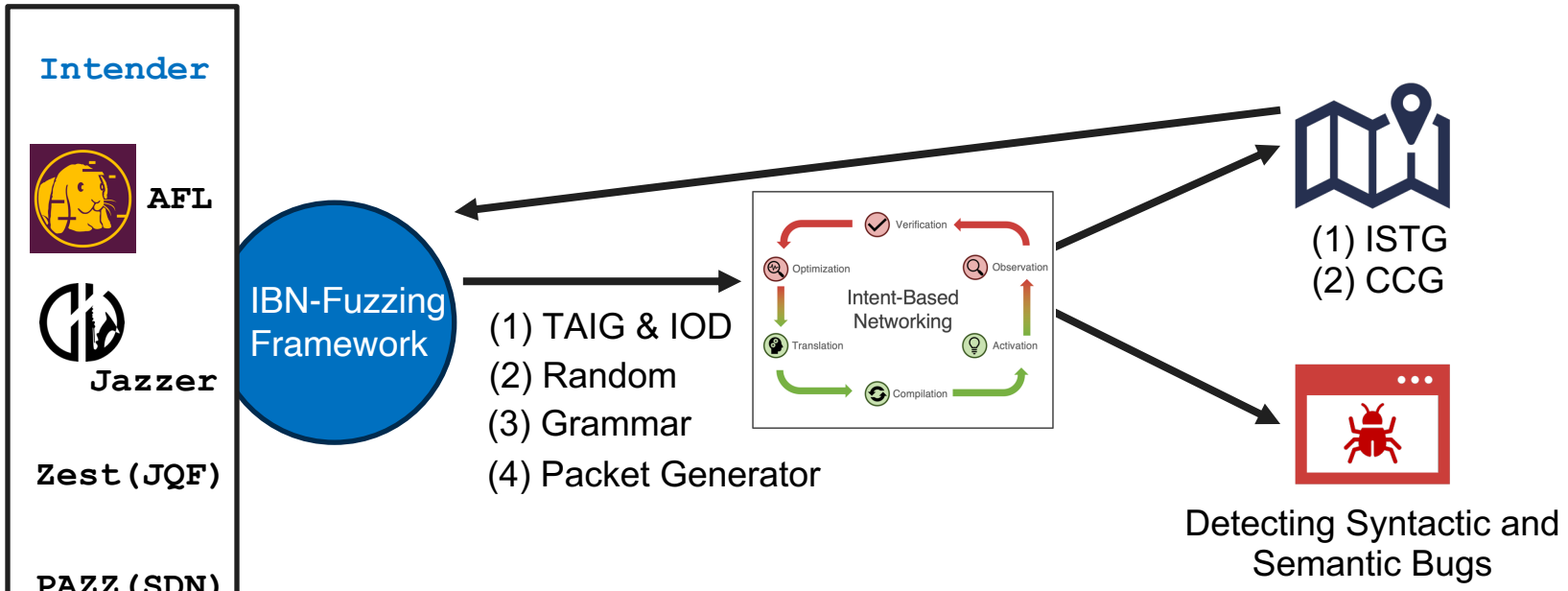


Detecting Connection Failure [3/3]

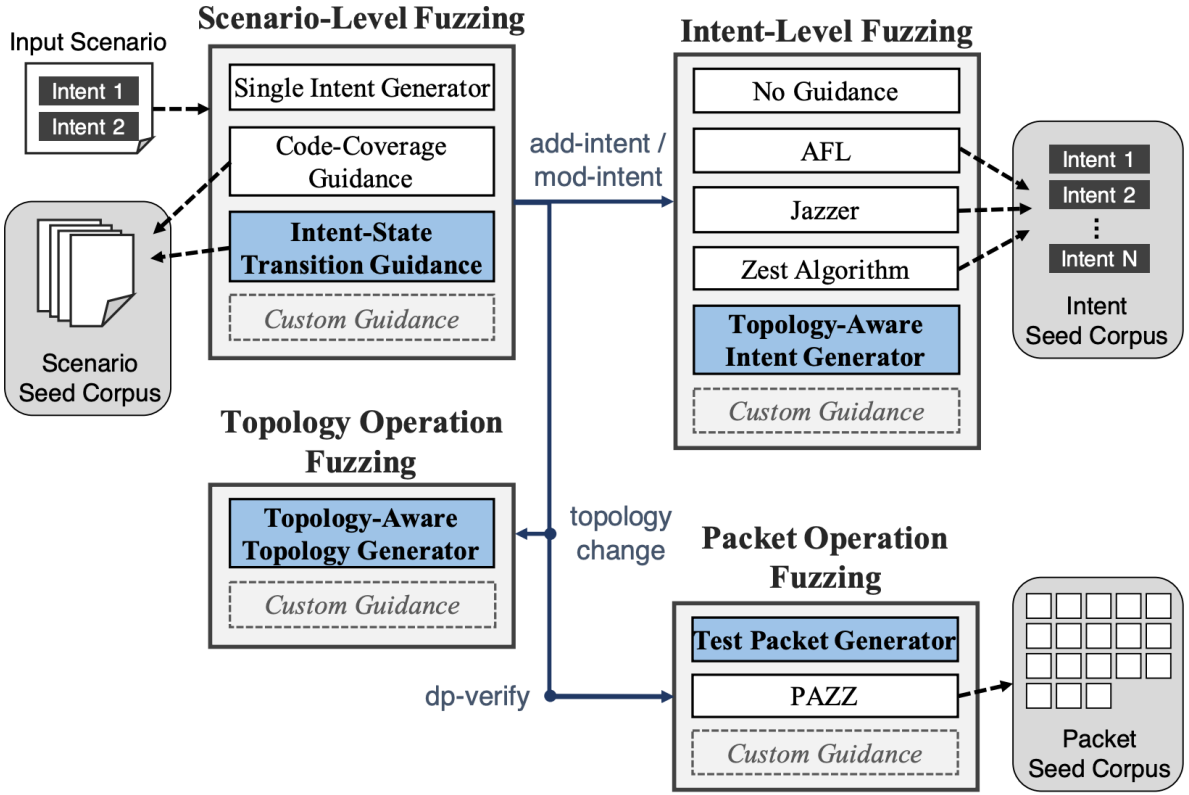


1. Point-interleaving switches deliver packets.
2. Point-interleaving hosts sends or sniffs packets.

Intender as a Framework



Intender as a Framework: Multi-Layer Fuzzing



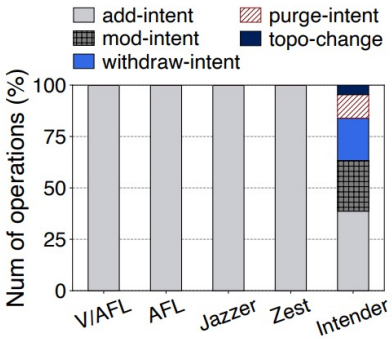
Evaluation [1/4]

- Environment Setup
  - Google Cloud VM: 4 vCPU, 16GB MEM, 60GB SSD
  - ONOS v2.5.1
- Found **12 new bugs (11 security-critical CVEs)**
  - 9 semantic bugs
  - Security impacts: network-wide **denial of service** & **tampering**
- Compare 4 existing fuzzers (AFL, Jazzer, Zest, PAZZ)
  - Up to **2.2×** better in branch coverage
  - Up to **82.6×** more number of unique errors

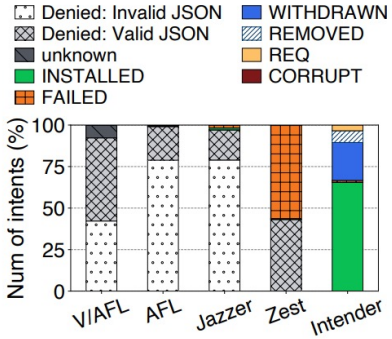
Evaluation [2/4]

- Improve fuzzing performance compared to baselines
  - *Topology-Aware Input Generation* (TAIG) can produce **78.7×** more valid intents
  - *Intent-Operation Dependency* (IOD) can reduce **73.02%** of redundant operations
  - *Intent-State Transition Guidance* (ISTG) leads to **1.8×** more intent-state transitions than code coverage guidance (CCG)

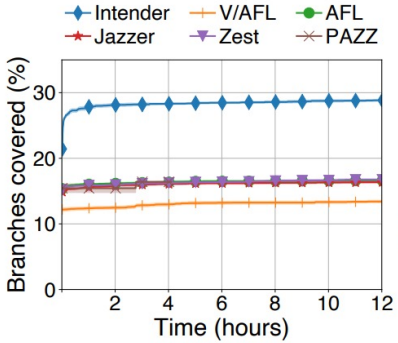
Evaluation [3/4]



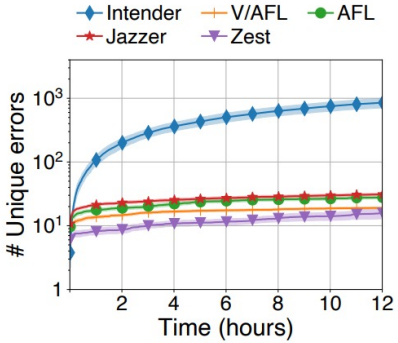
(a) Number of operations.



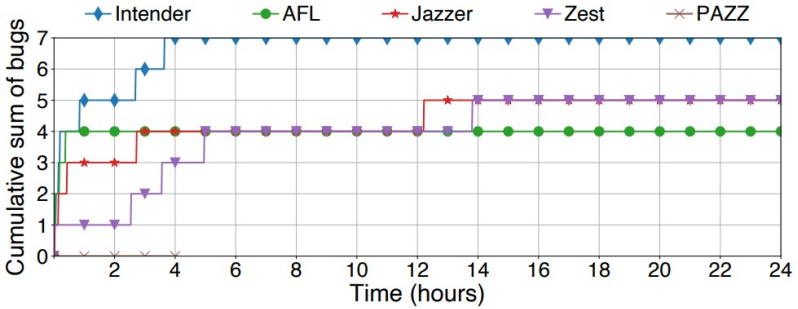
(b) Number of intent states.



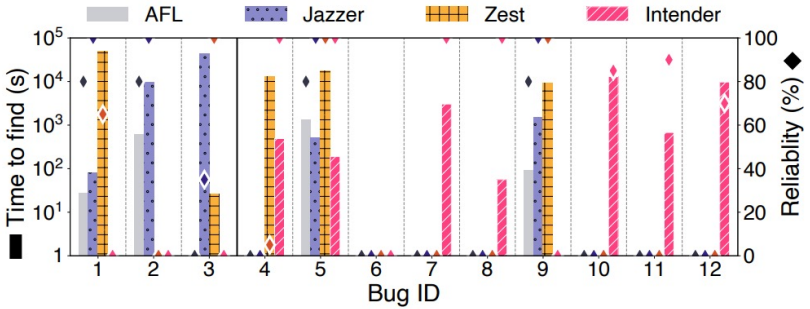
(c) Branch coverage.



(d) Number of unique errors.

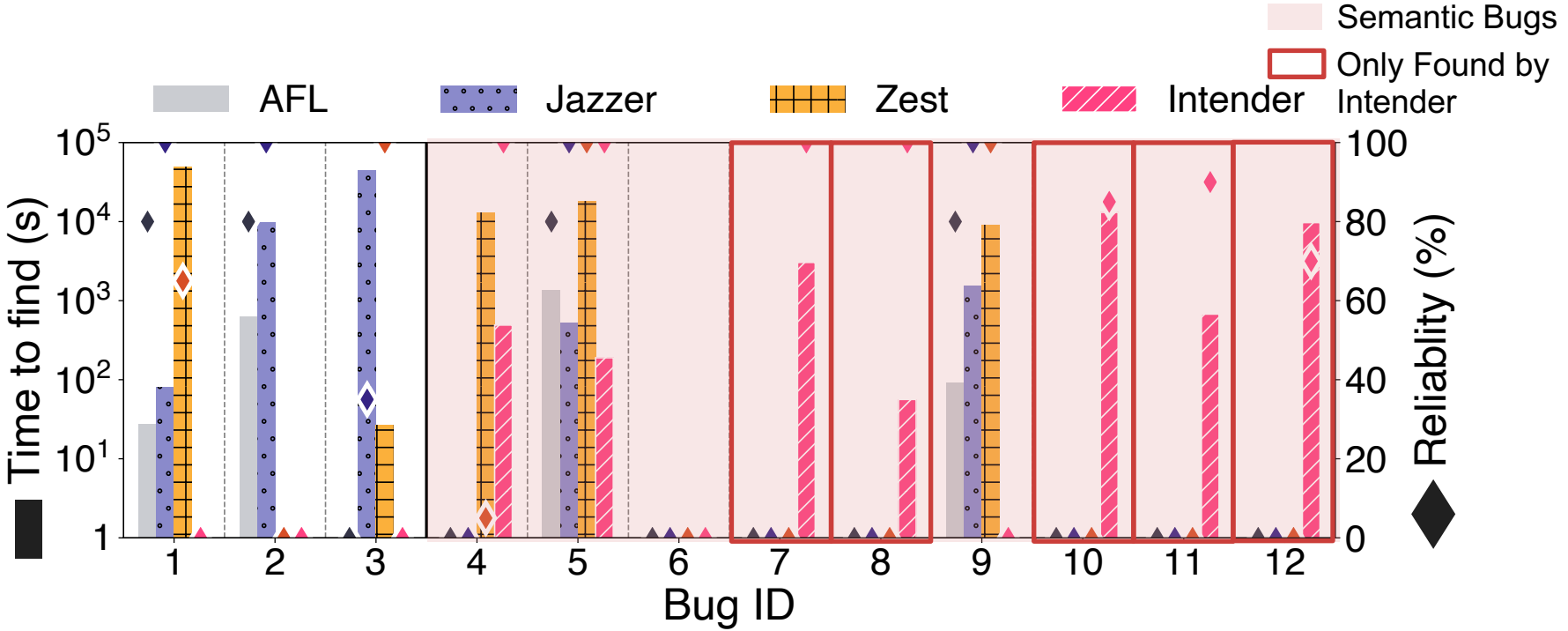


(e) Cumulative sum of bugs found.



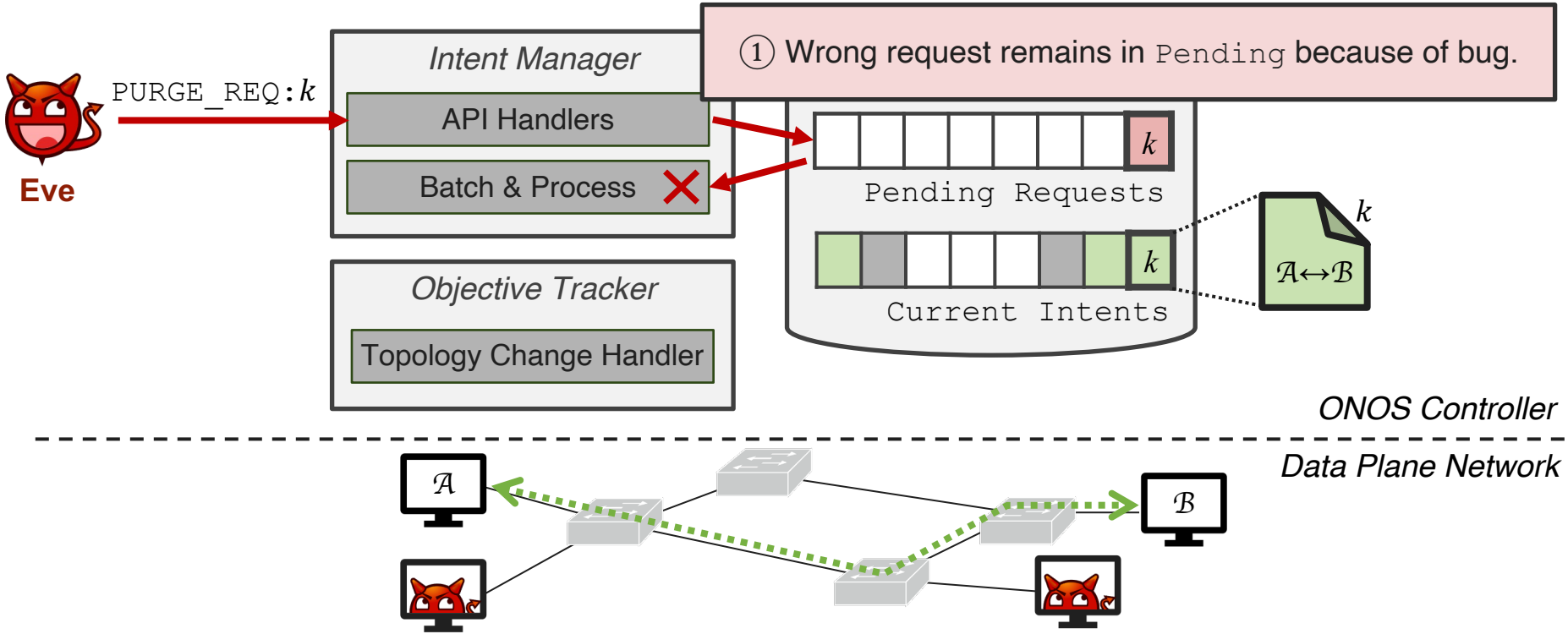
(f) Mean time to find each bug.

Evaluation: Time-to-find Bug Efficiency [4/4]



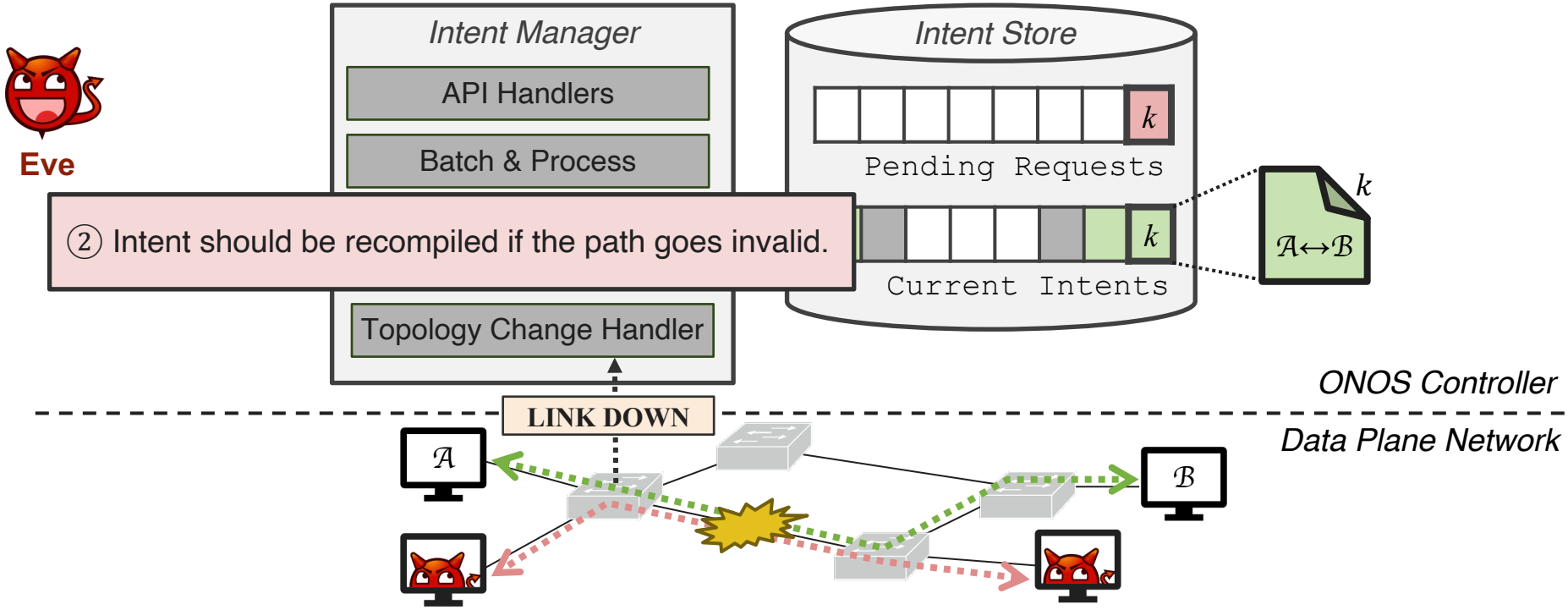
Case Study: CVE-2022-24035

(1) **Eve** requests `PURGE` on `INSTALLED` intent



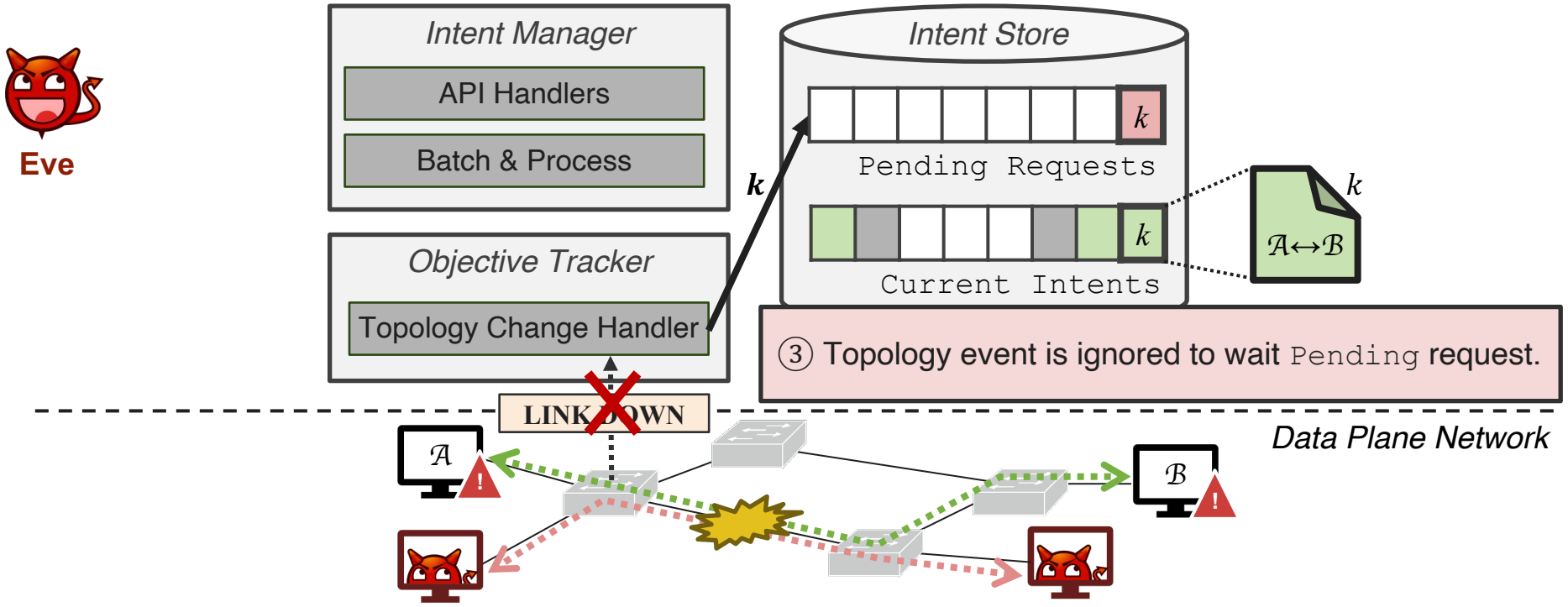
Case Study: CVE-2022-24035

(2) **Eve** exploits link-flooding attack



Case Study: CVE-2022-24035

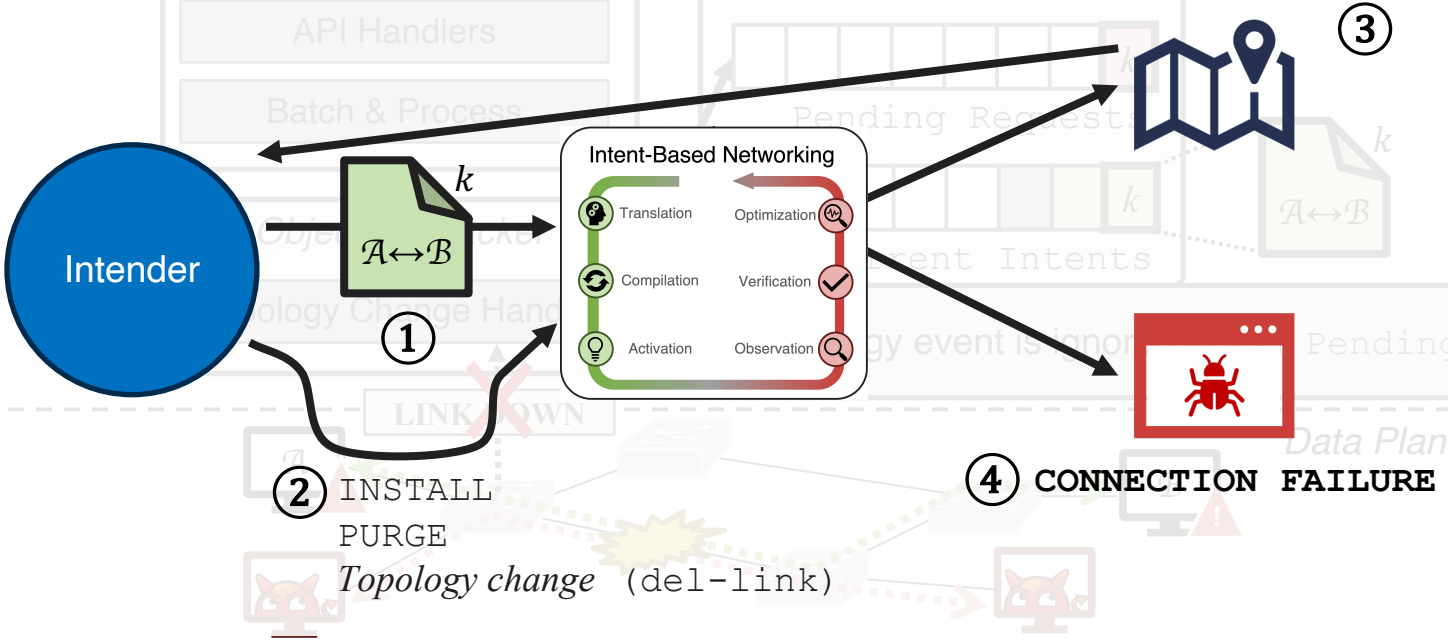
(3) Intent **DOES NOT** respond to topology event any more → **DoS**



Case Study: CVE-2022-24035

- *To discover this bug efficiently…*

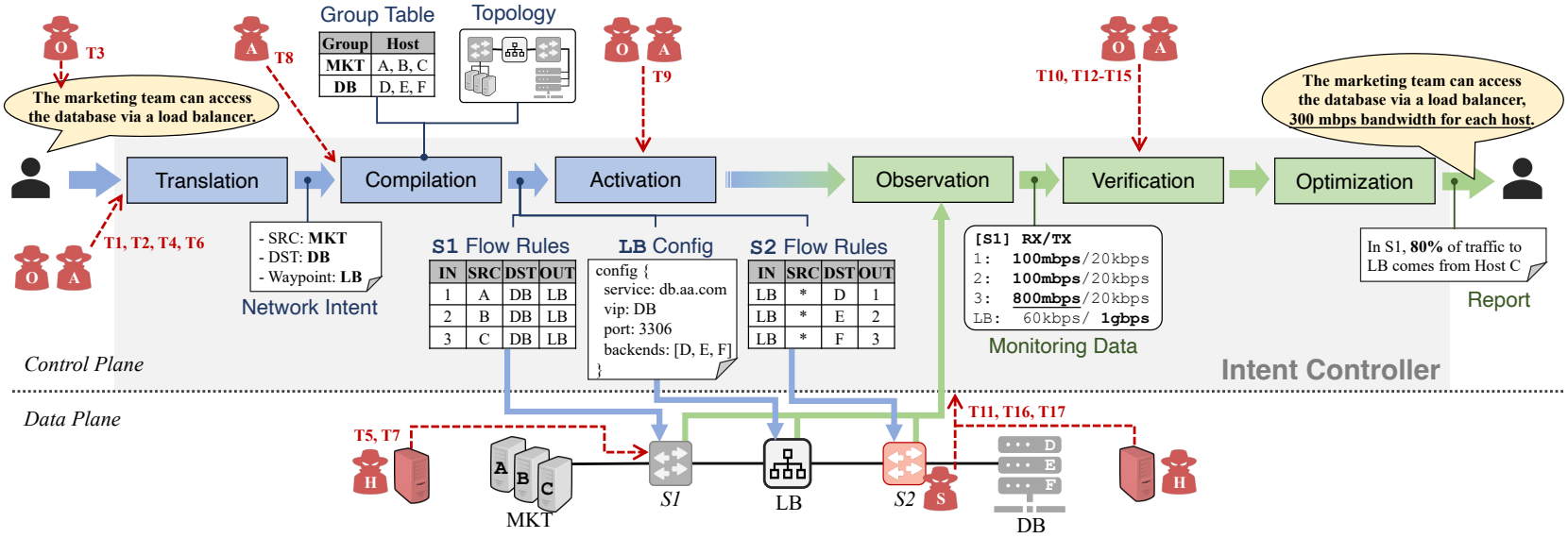
Intent Transition History			
INSTALL	$\mathcal{I}$		
PURGE	$\mathcal{I}$		
del-link	$\mathcal{I}$		



Conclusions of Intender

- Analyzed **186 bugs** in ONOS IBN
- Designed new fuzzing techniques for IBN
  - *Topology-Aware Intent Generation* (**TAIG**)
  - *Intent-Operation Dependency* (**IOD**)
  - *Intent-State Transition Guidance* (**ISTG**)
- Developed Intender architecture with 4 fuzzers
  - AFL, Jazzer, Zest, PAZZ
- Found **12 new bugs (11 CVEs)** in ONOS IBN

Future Research on IBN Security



Thank you!

Questions: kim1685@purdue.edu

Backup Slides

Found Bugs

Semantic Bugs

Only Found by Intender

#	CVE ID	Type	Operation	Description
1	CVE-2021-38363	**SYN2**	`add-intent`	`PointToPoint` intent with invalid point field causes NullPointerException
2	CVE-2022-29604	**SYN4**	`add-intent`	`PointToPoint` intent which has an upper-case letter in a device ID shows `CORRUPT`
3	CVE-2022-29606	**SYN4**	`add-intent`	`PointToPoint` intent which has a large switch port number shows `CORRUPT`
4	CVE-2022-29609	**SEM1**	`add-intent`	`HostToHost` intent with the same source and destination shows `INSTALLING`
5	CVE-2022-29608	**SEM2**	`add-intent`	`PointToPoint` intent installs an invalid flow rule causing network loop
6	CVE-2022-29605	**SEM2**	`add-intent`	Intent tries to install IPv6 flow rules into OF10 switches
7	CVE-2022-29944	**SEM3**	`add-intent`	Intent cannot bypass intents with higher priority
8	CVE-2021-38364	**SEM3**	`add-intent`	Intent can delete or modify flow rules of previous intents which share the path
9	-	**SEM4**	`add-intent`	`PointToPoint` intent with switch port 0 installs useless flow rules
10	CVE-2022-24109	**SEM3**	`withdraw-intent`	Deletion of one of the duplicate intents removes all flow rules
11	CVE-2022-29607	**SEM1**	`mod-intent`	`HostToHost` intent modified to have same source and destination shows `INSTALLED` without any flow rules
12	CVE-2022-24035	**SEM1**	`purge-intent` & `topology-change`	After requesting purge on installed `PointToPoint` intent, the state of intent does not change to `FAILED` with link failure