Source Routing Based In-band Network Telemetry

不時輕聲地以 Source Routing 探測網路的鄰座艾莉同學

Introduction

01

Problem to Solve

Whether the flow rules populated by the controller are strictly executed by the data plane.

02

Goal

Verify the incosistency between data plane and control plane in P4 switch

Related work

Fuzzing-based testing

- Try to use fuzzer to exploit the switch
- Time consumming!

Symbolic execution-based testing

- Use SMT solver to generate probing packet satisfying all constraints.
- Try to cover all rules and paths.
- Time consuming!
- Need a lot of probing packet

However, we should pay more attention on

01 Key nodes with vital network function

e.g., firewall

02 Critical links passing through these nodes

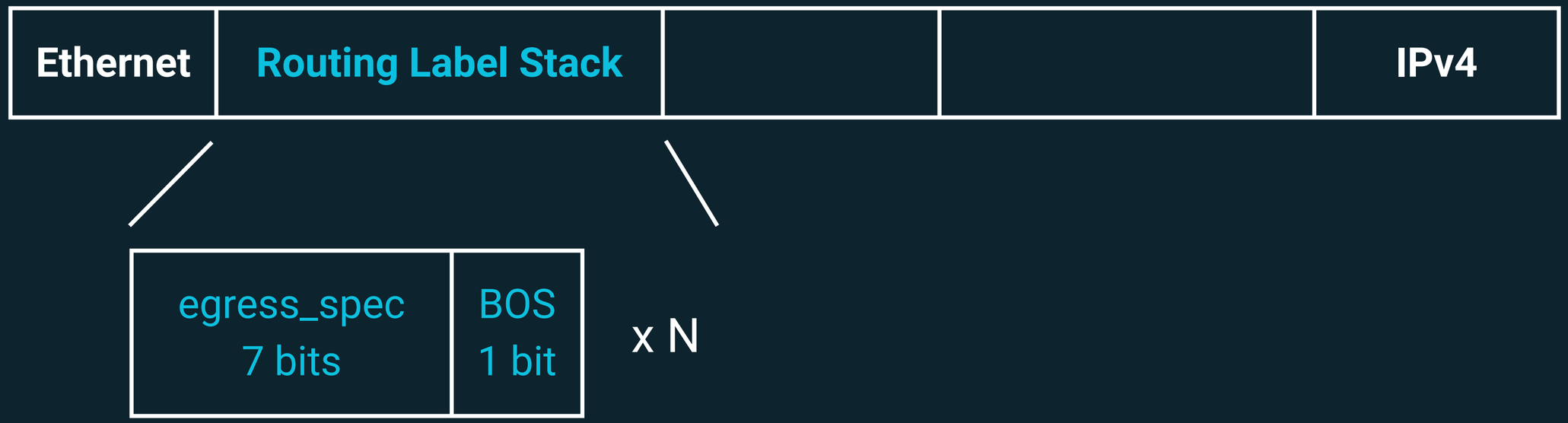
Design of our work

Refer to:
J. Xia, P. Cui, Z. Li and J. Lan, "SRCV: A Source Routing based Consistency Verification Mechanism in SDN," 2021 3rd International Conference on Advances in Computer Technology, Information Science and Communication (CTISC), Shanghai, China, 2021, pp. 77-81

Source Routing

In-band Network Telemetry

Design of Packet Format



Source Routing

Design of Packet Format

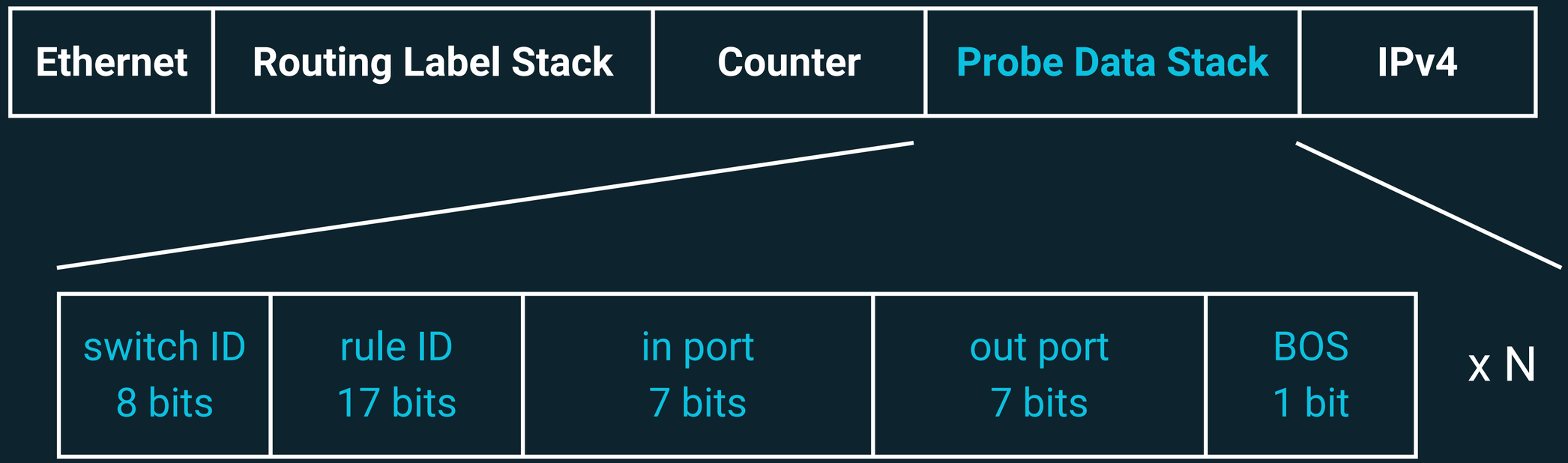
visited_count

Store how many hops the probing packet has visited.

8 bits

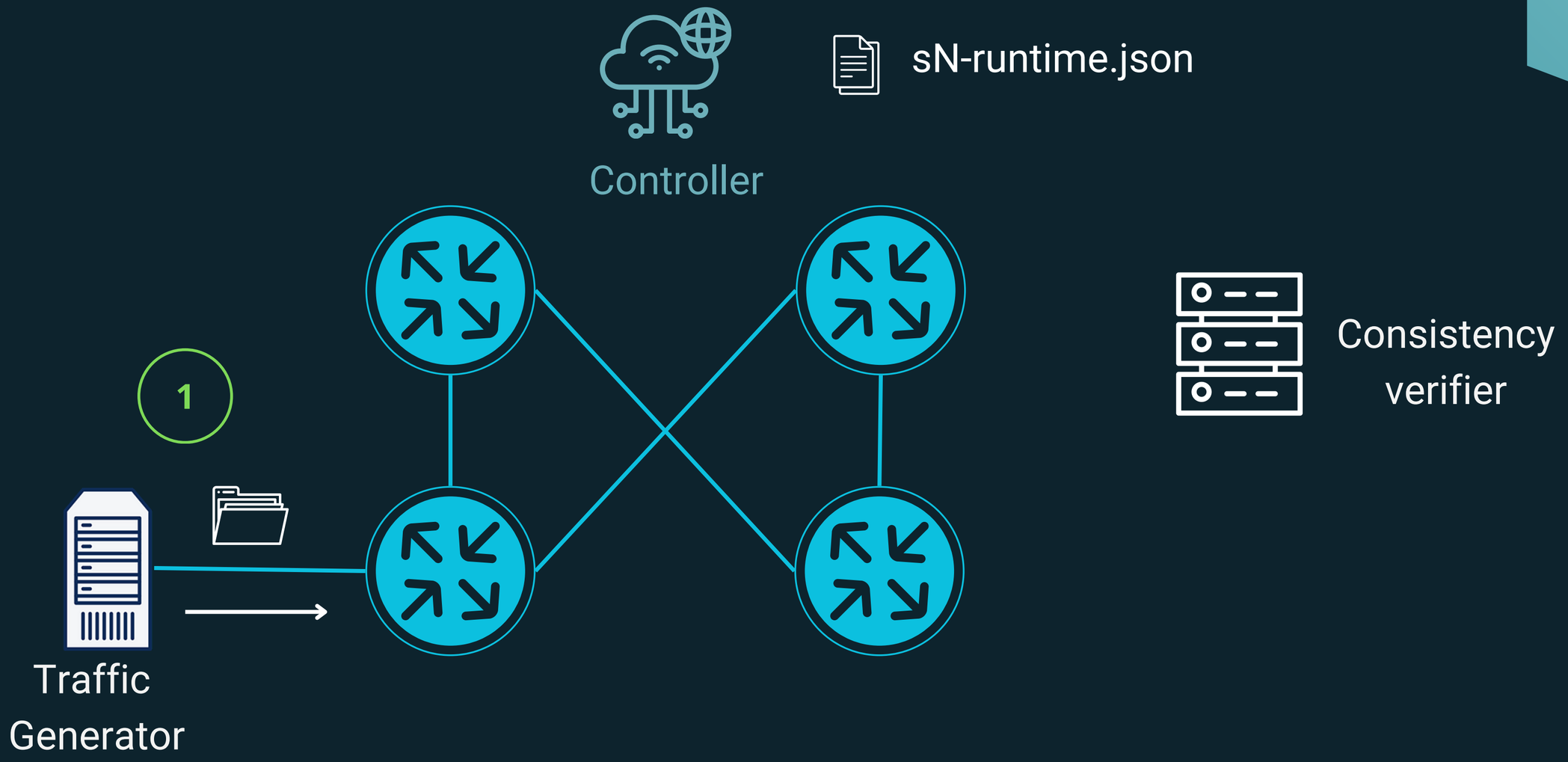
In-band Network Telemetry

Design of Packet Format

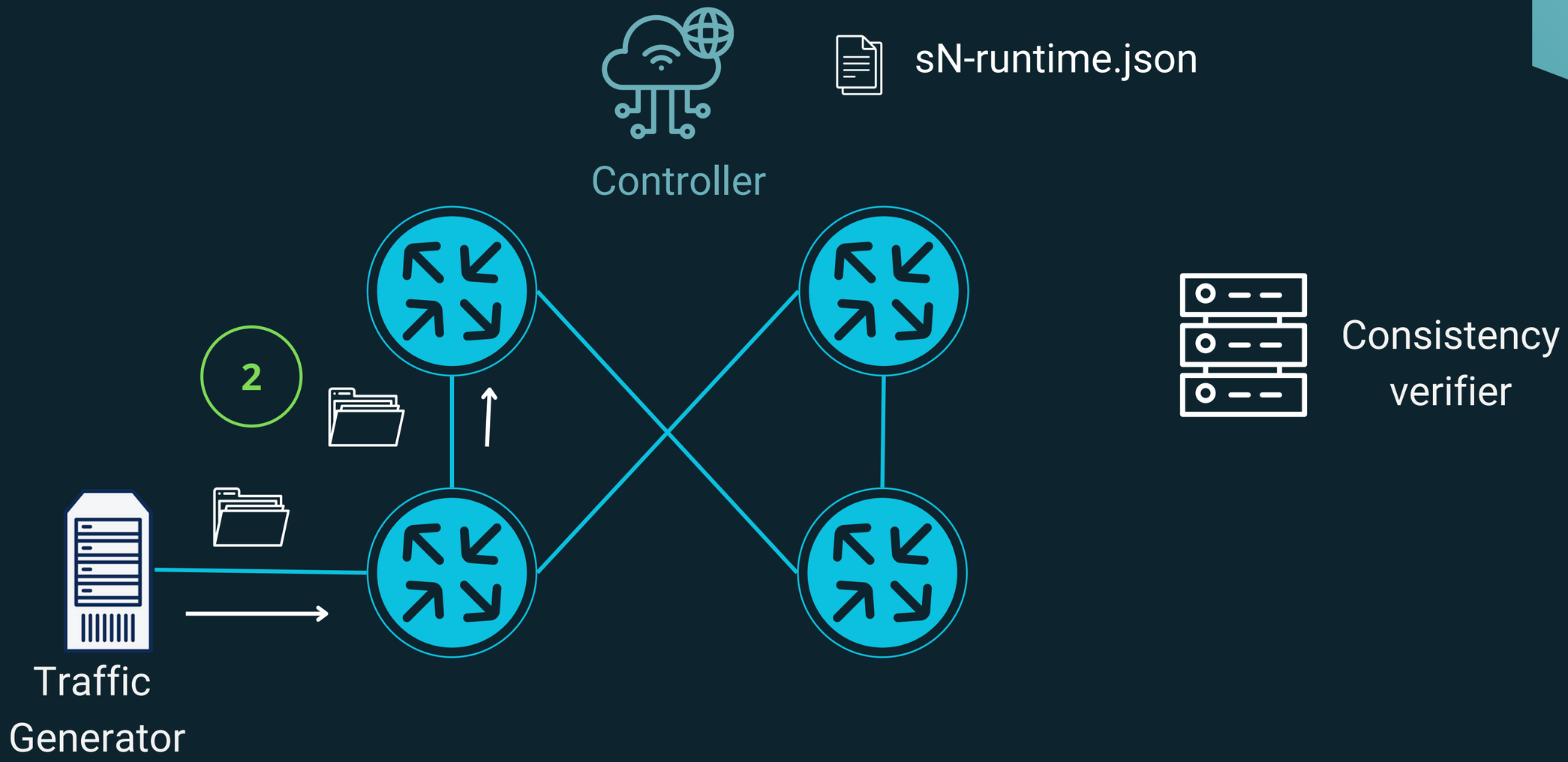


In-band Network Telemetry

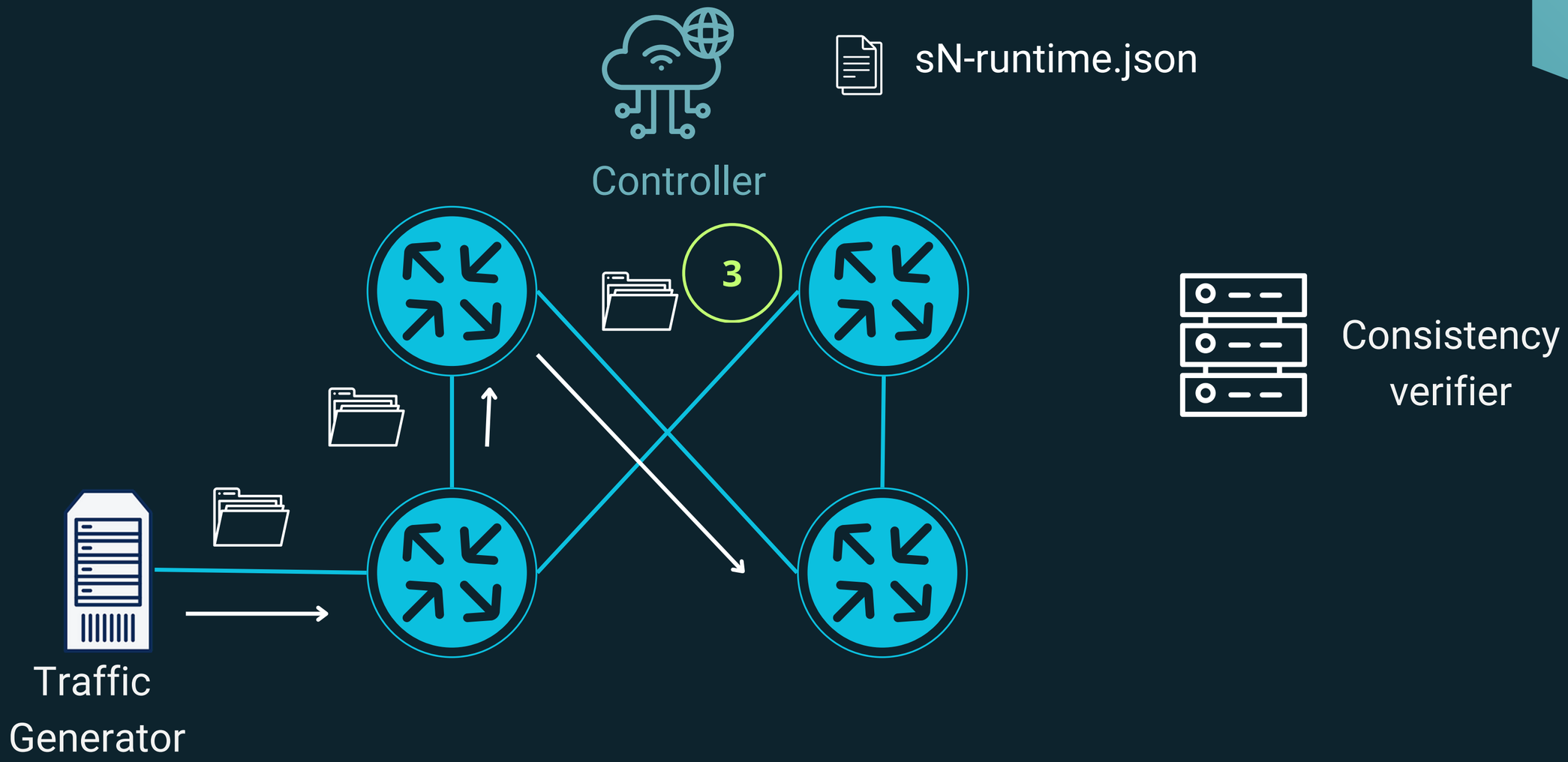
System Architecture



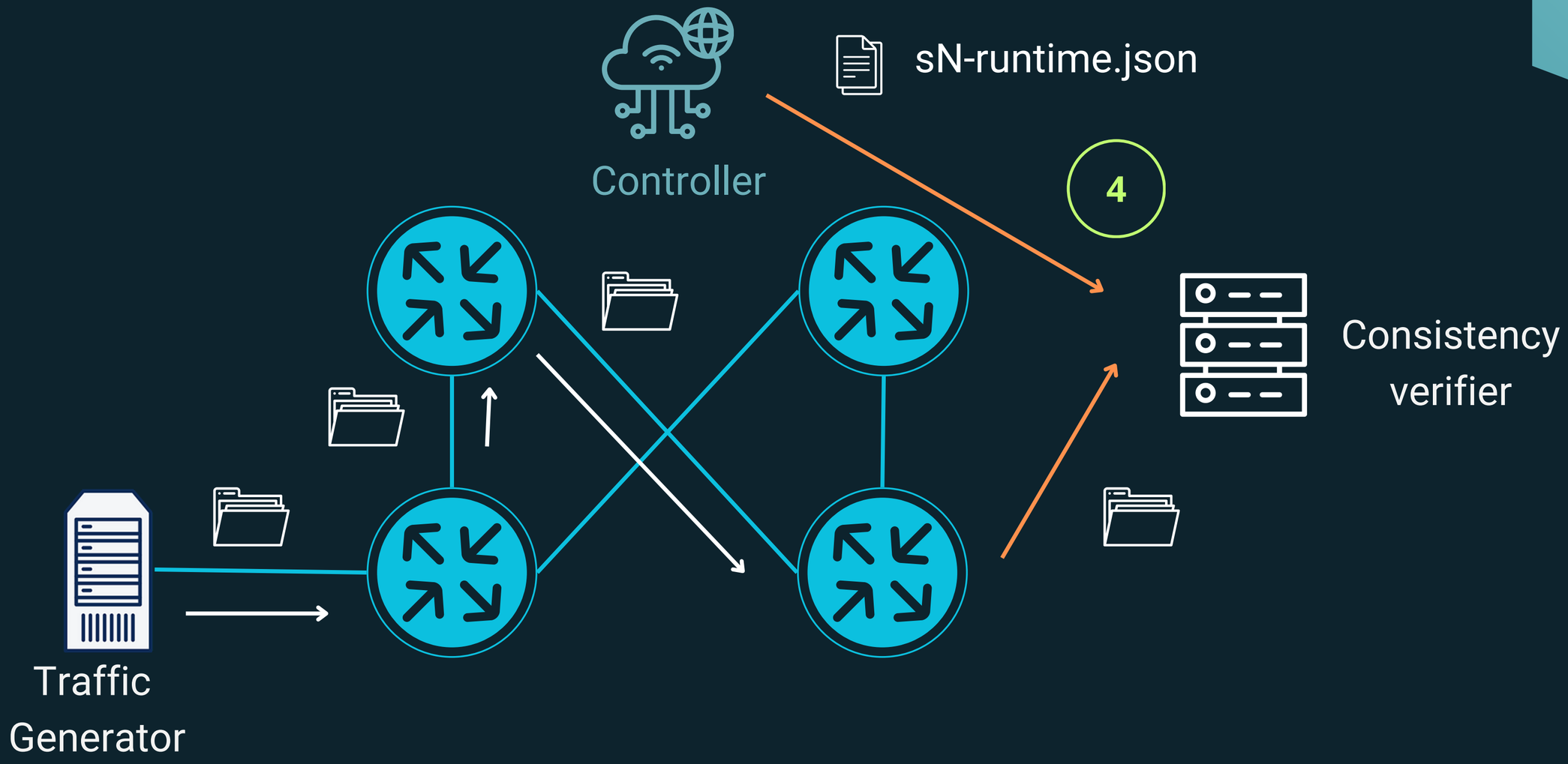
System Architecture



System Architecture



System Architecture



Consistency Verification Algorithm

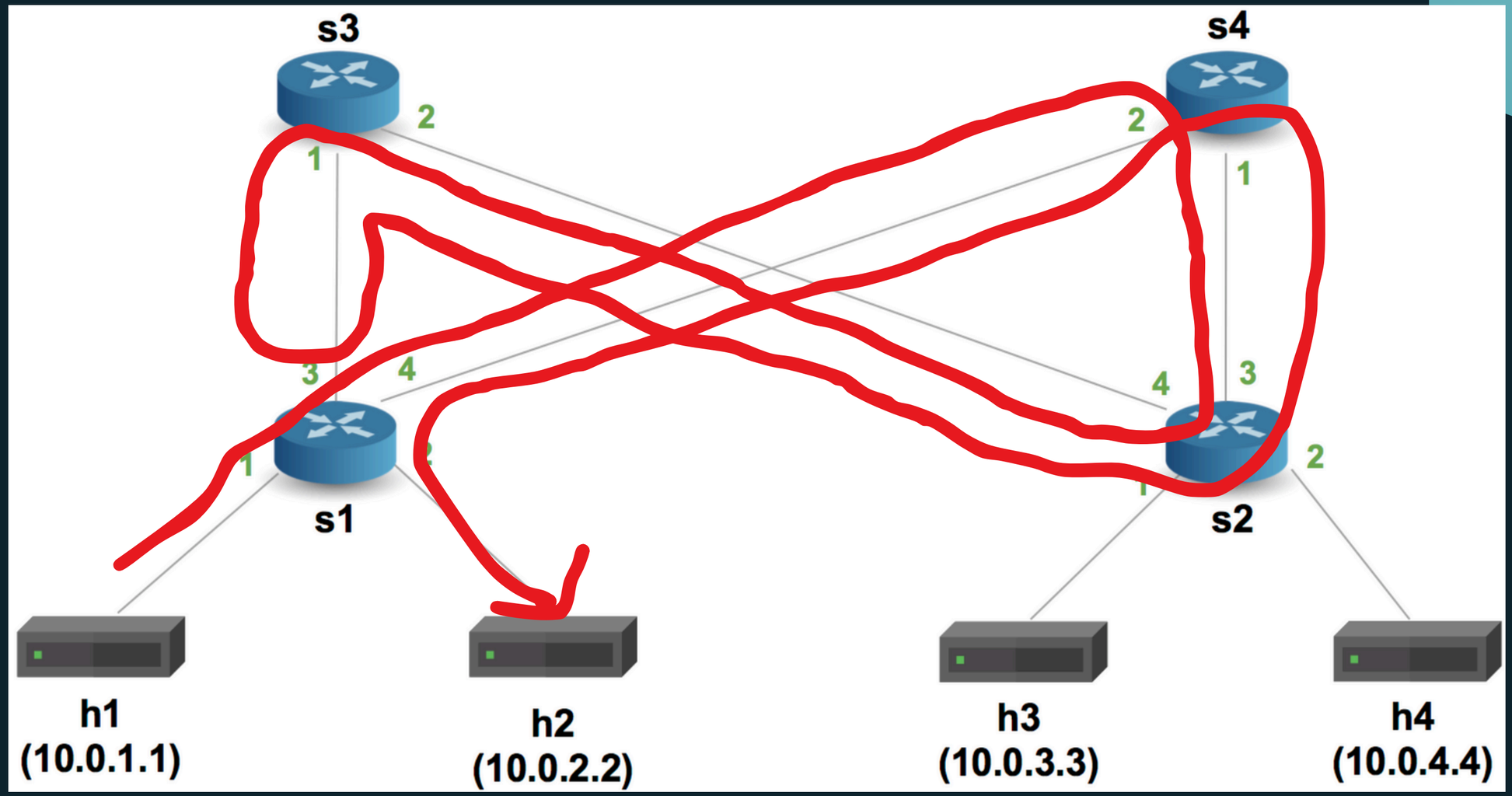
Input: The network configuration ***Rules***, the symbolic packet ***SP***, and the given verification path ***PATH_SPEC***.

Output: The checking results ***PATH_CHECK*** and the ***Error_Report*** for inconsistent switches.

```
1:  for switch ∈ PATH_SPEC do
2:    if (last switch) then
3:      for rule ∈ Rules do
4:        if check (SP, rule) == TRUE then
5:          PATH_CHECK ← TRUE // no faults in this given path
6:        else
7:          if (last rule) then
8:            SWITCH_CHECK ← False
9:            PATH_CHECK ← False   // inconsistent path
10:           Error_Report
11:   else
12:     for rule ∈ Rules do
13:       if check (SP, rule) == TRUE then
14:         SWITCH_CHECK ← TRUE // consistent switch
15:         Go to next switch // continue checking the next node
16:       else
17:         if (last rule) then
18:           SWITCH_CHECK ← False
19:           PATH_CHECK ← False // inconsistent path
20:           Error_Report
```

Demo

Topology



routing sequence: 4 1 4 1 3 2 3 2 2

分工表

陳泓文 r13922061	literature collection and review, P4 implementation, Slide, Demo recording
葉富銘 r13922146	sender implementation, introduction and related work of report
楊東翰 r13922074	Consistency Verification Implementation, Demo recording
黃恩明 r13922078	Receiver Implementation / Report

Thanks for Listening!

模板

The Rise of Artificial Intelligence in Cybersecurity

Automation

AI can automate repetitive tasks, freeing up security professionals to focus on more strategic initiatives.

Intelligence

AI algorithms can learn from data and identify patterns that humans might miss. This can help to detect threats earlier and prevent attacks.

Adaptability

AI can adapt to changing threats and environments, making it more effective than traditional security solutions.

Machine Learning for Threat Detection and Response

01

Anomaly Detection

Machine learning algorithms can identify unusual patterns in network traffic, user behavior, and other data, which may indicate a cyber attack.

02

Behavioral Analysis

ML can analyze user behavior and identify potential threats based on changes in their actions, such as unusual login attempts or file downloads.

03

Threat Intelligence

Machine learning can be used to analyze threat intelligence feeds, identify emerging threats, and update security systems.

Predictive Analytics: Anticipating Cyber Attacks

Data Analysis

AI analyzes historical data to identify patterns and trends that indicate potential future attacks.

Risk Assessment

AI models assess the likelihood of different threats based on the identified patterns and vulnerabilities.

Proactive Measures

Security teams can take preventative measures to mitigate identified risks before they become actual threats.

Automated Incident Response and Remediation

Network Segmentation

AI can segment networks to limit the damage caused by an attack and prevent attackers from gaining access to critical systems.

Vulnerability Patching

AI can automatically patch vulnerabilities and update security systems to reduce the risk of attacks.

Data Recovery

AI can help to recover data that has been lost or corrupted due to a cyber attack.

Threat Isolation

AI can automatically isolate infected systems to prevent the spread of malware and protect sensitive data.

Enhancing Security with Natural Language Processing

01

Threat Intelligence

NLP can analyze threat intelligence feeds to identify emerging threats and vulnerabilities.

02

Phishing Detection

NLP can detect phishing emails and other social engineering attacks by analyzing the content and identifying suspicious patterns.

03

User Authentication

NLP can be used to verify user identity through voice or text-based authentication.

Securing the Internet of Things with AI-Driven Solutions

Device Security

AI can help to secure IoT devices by identifying and mitigating vulnerabilities.

Network Protection

AI can protect IoT networks from attacks by detecting and preventing malicious activity.

Data Privacy

AI can help to protect the privacy of data generated by IoT devices.

The Future of AI-Powered Cybersecurity: Trends and Innovations

Advanced Threat Detection

AI will continue to evolve and become more effective at detecting and responding to threats.

Automated Security Operations

AI will automate more security tasks, making security operations more efficient and effective.

Ethical AI in Cybersecurity

The responsible use of AI in cybersecurity is crucial to ensure that it is used to protect people and systems.