

BUZZ: TESTING CONTEXT-DEPENDENT POLICIES IN STATEFUL NETWORKS

-INTURI SAI HARSHINI

1

CONTENTS

- INTRODUCTION
- WHY IS IT CHALLENGING?
- WORKFLOW OF BUZZ
- IMPLEMENTATION
- EVALUATION

INTRODUCTION

- Checking whether a network correctly implements intended policies is challenging.
- Can X talk to Y?
- Existing approaches face fundamental expressiveness and scalability challenge.
- Current abstractions cannot capture stateful behaviors.
- Trying to reason about stateful behaviors results in state space explosion.

WHY IS IT CHALLENGING?

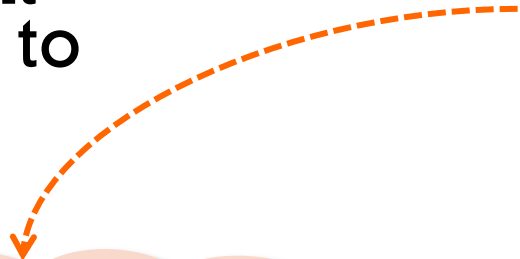
Does the network do what I want it to

do?

???

What I want
the network to
do

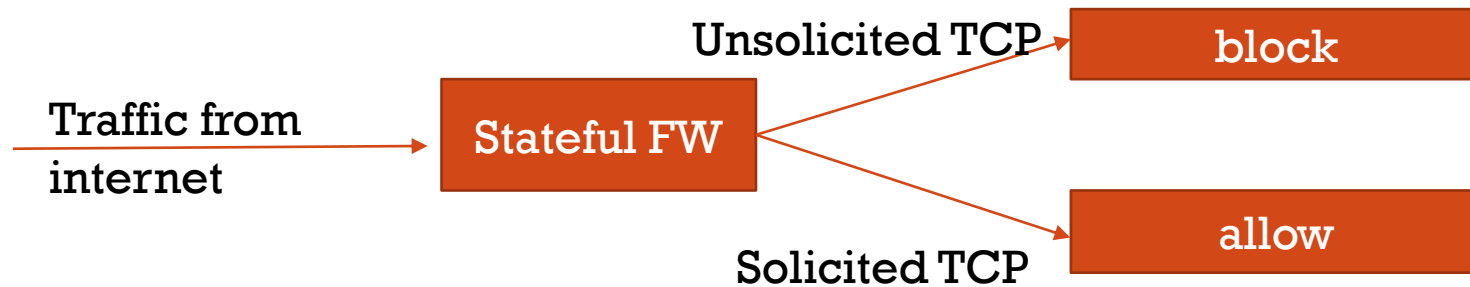
 Network
operator



Reality
What the network
does
network

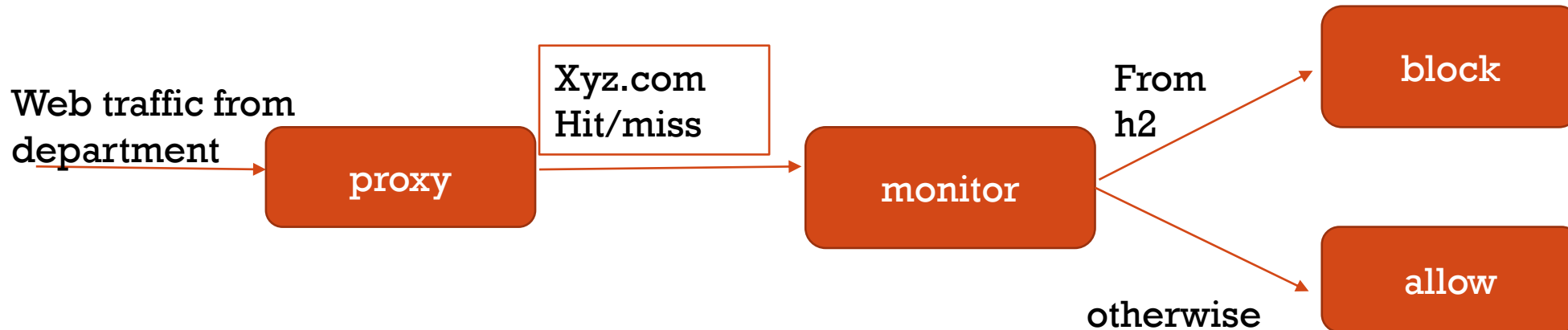


STATEFUL FIREWALLING



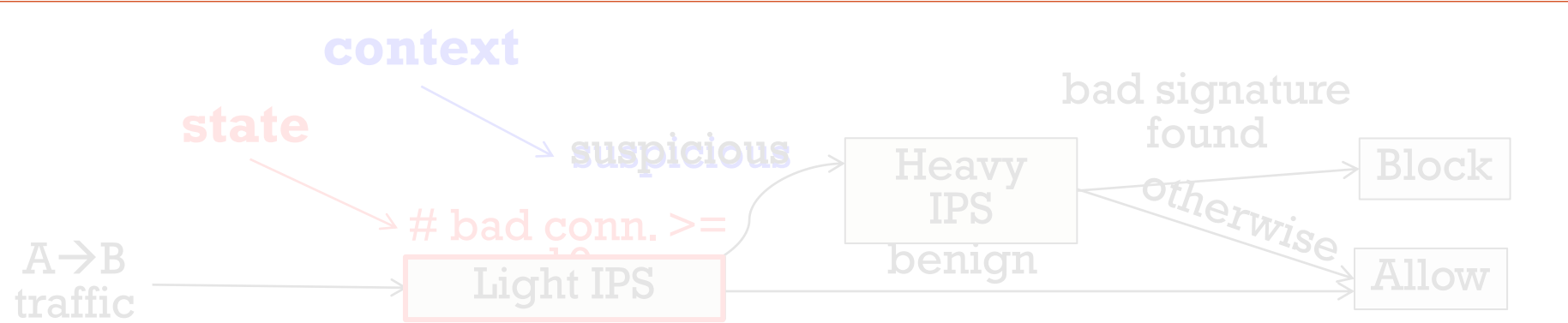
- Incoming traffic is allowed depending on its context.
- Even this simple policy cannot be captured.
- It doesn't capture the policy-relevant state of the firewall.

CONTEXT-DEPENDENT TRAFFIC MONITORING



- Cache hits/misses for H2 should be monitored.
- There could be subtle policy violations
 1. The proxy hides traffic true origin.
 2. The proxy's response depends on the hidden policy-relevant state.

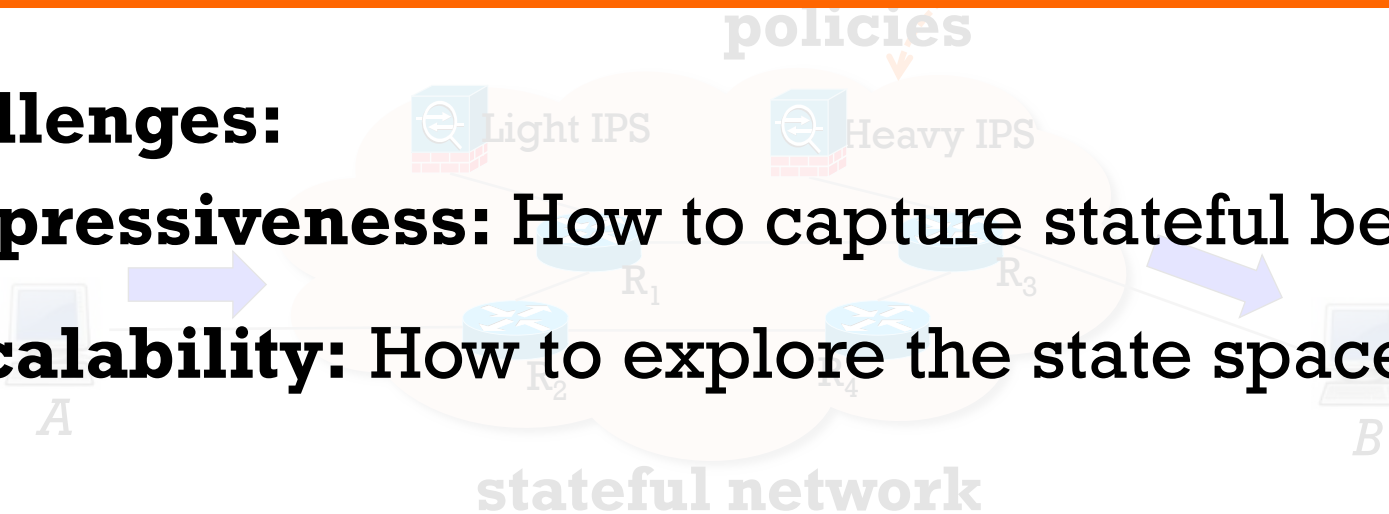
MULTI STAGE TRIGGERS



How can we check context-dependent policies in stateful networks?

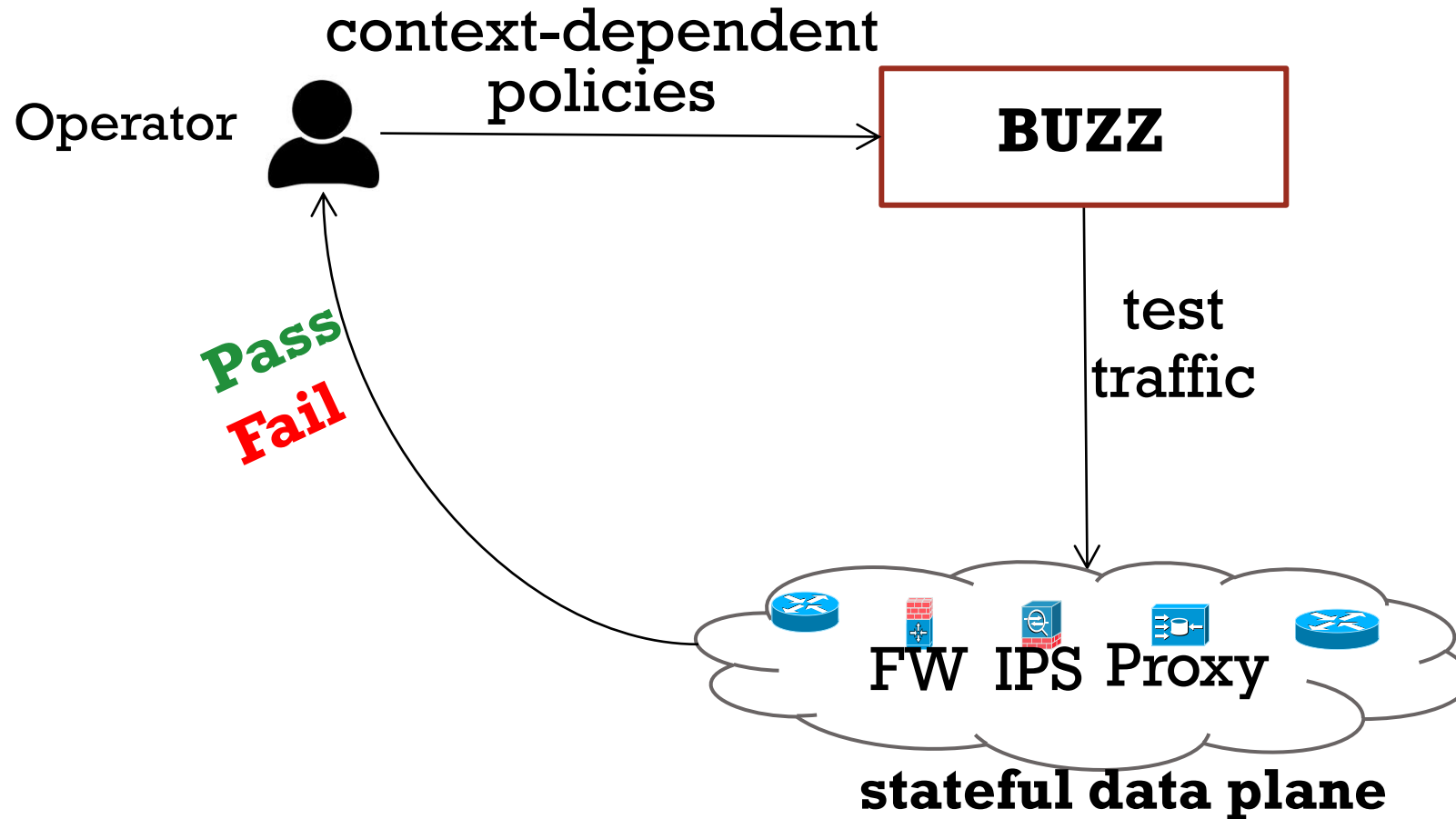
Challenges:

- **Expressiveness:** How to capture stateful behaviors?
- **Scalability:** How to explore the state space?

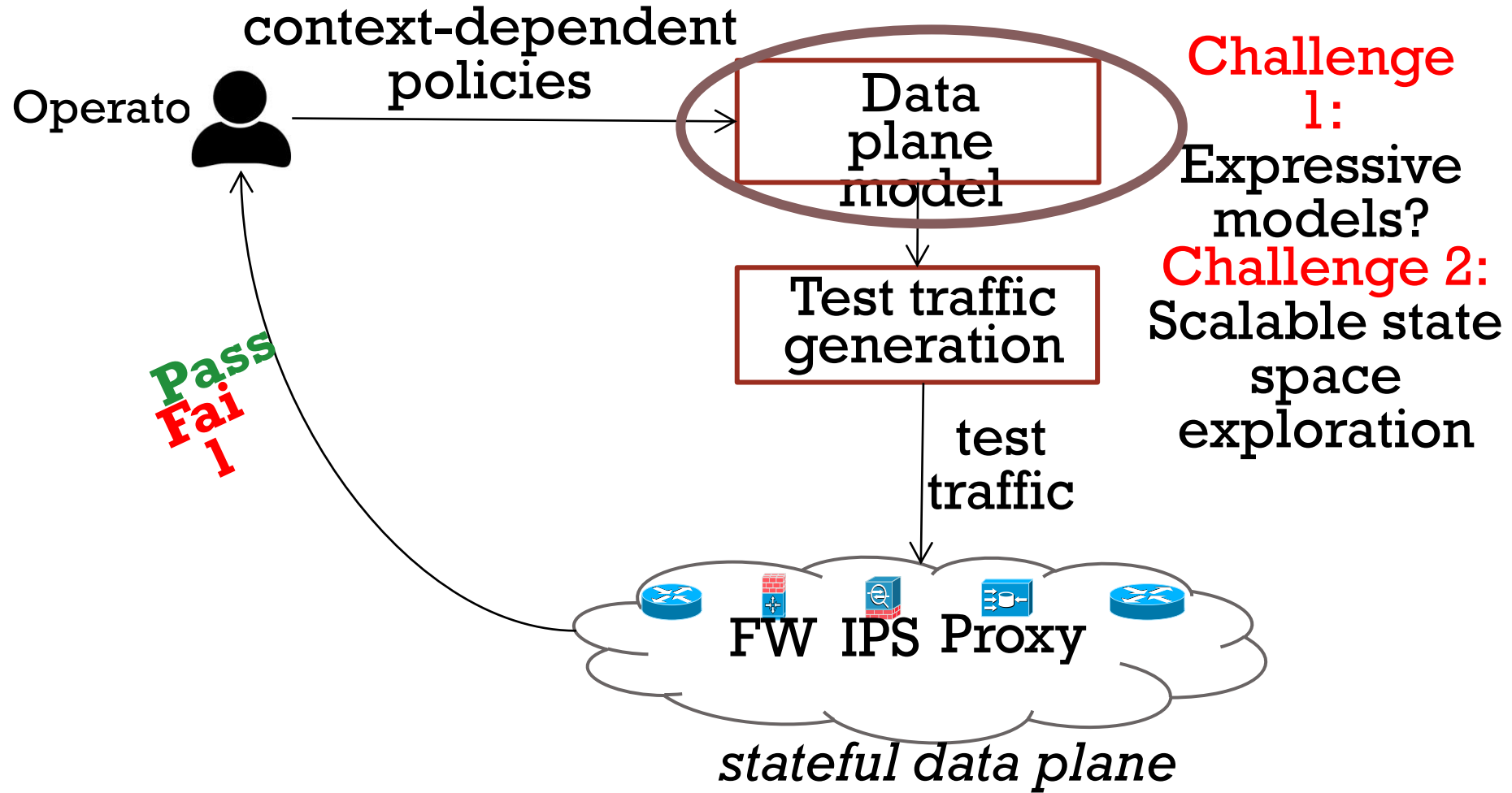


SOLUTION: BUZZ

BUZZ is an active testing framework to check context-dependent policies in stateful data planes

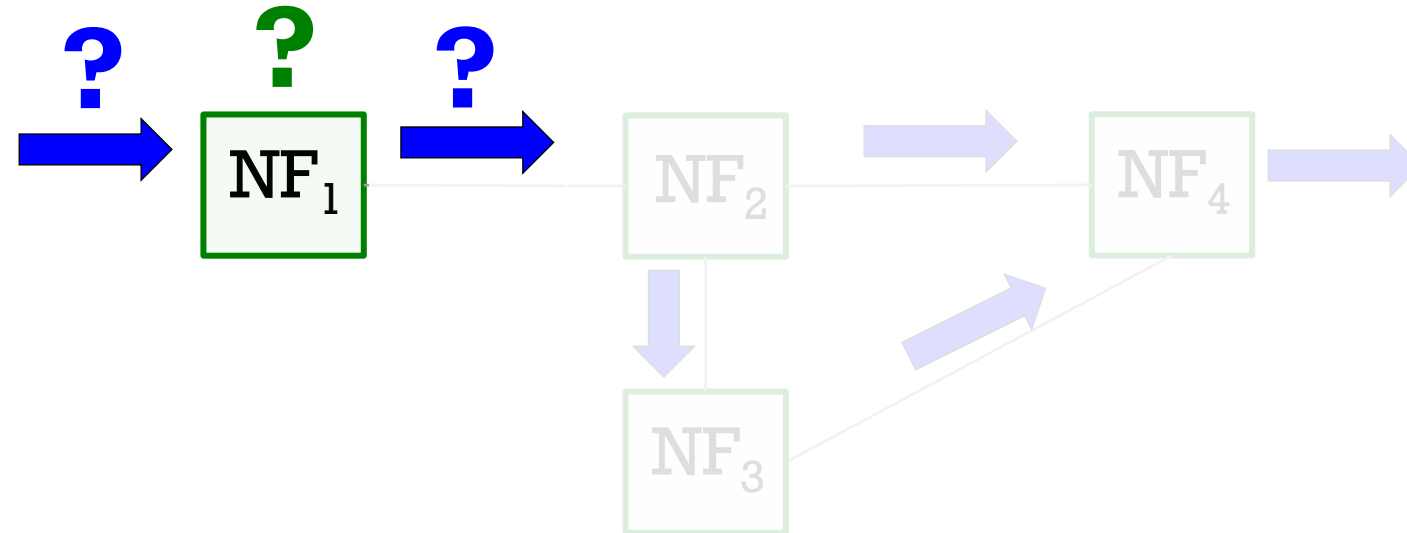


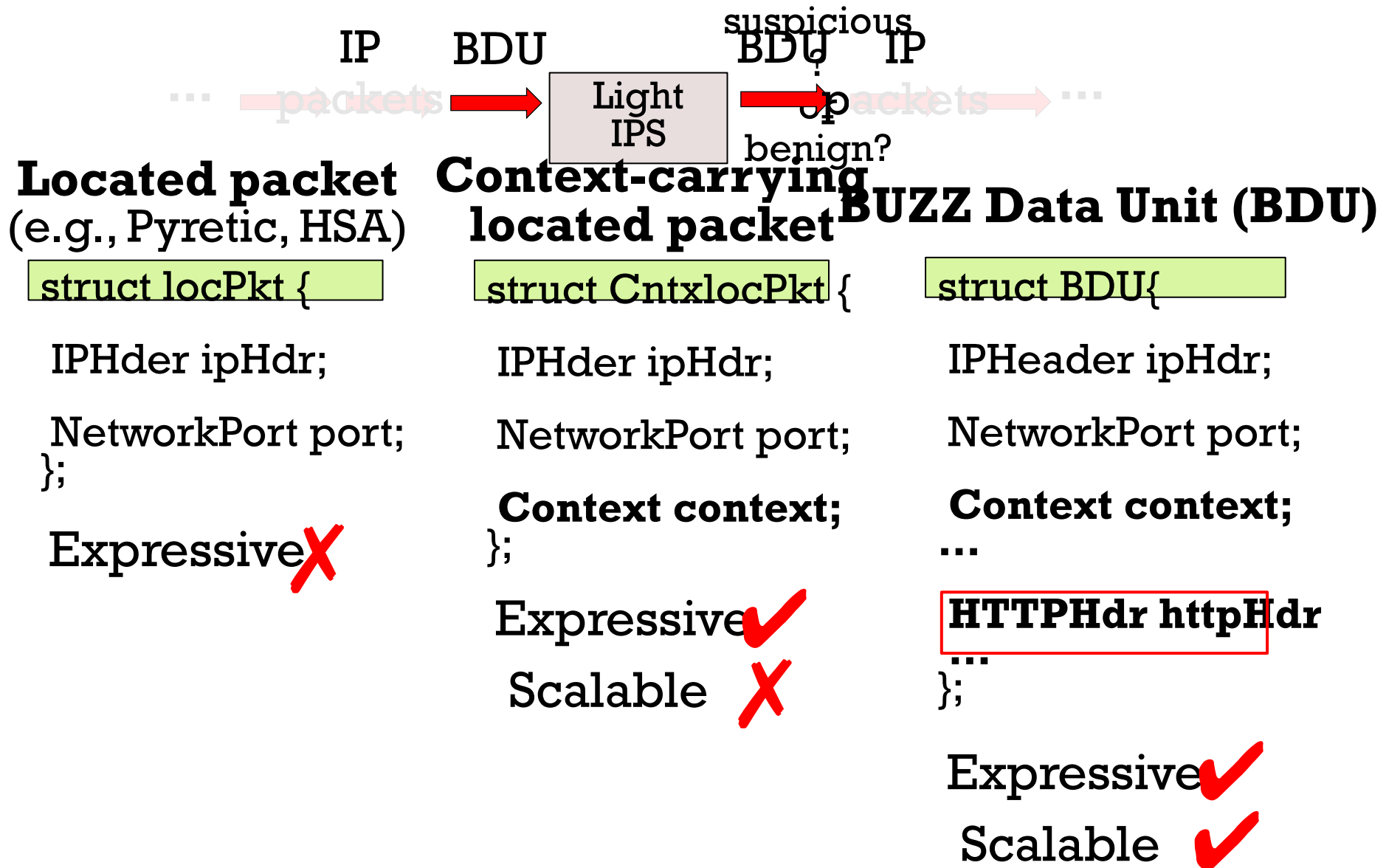
CHALLENGE 1: EXPRESSIVE DATA PLANE MODEL



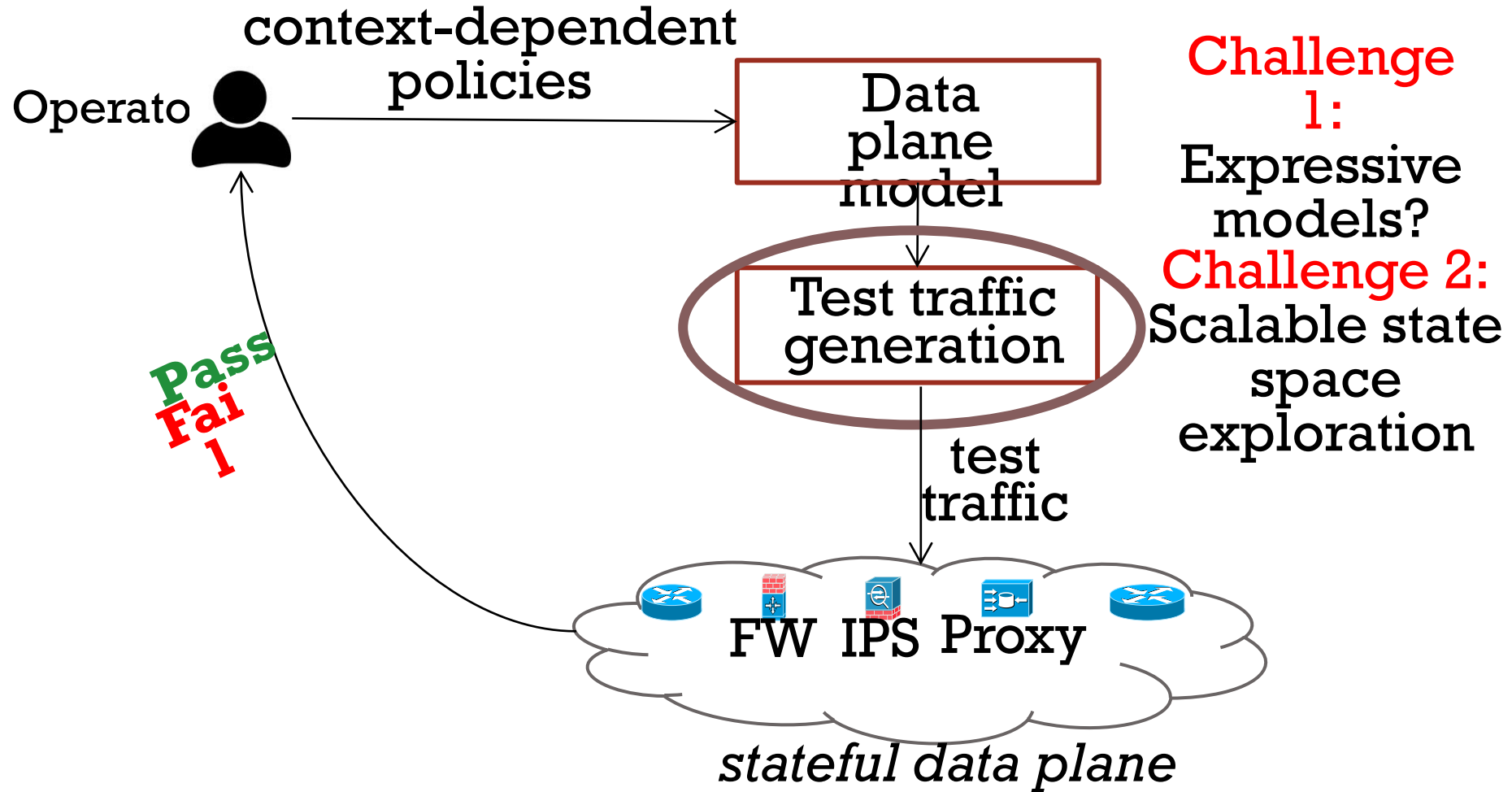
CHALLENGE 1: EXPRESSIVE DATA PLANE MODEL

1. How to model the traffic unit?
2. How to model a network function (e.g., an IPS)?

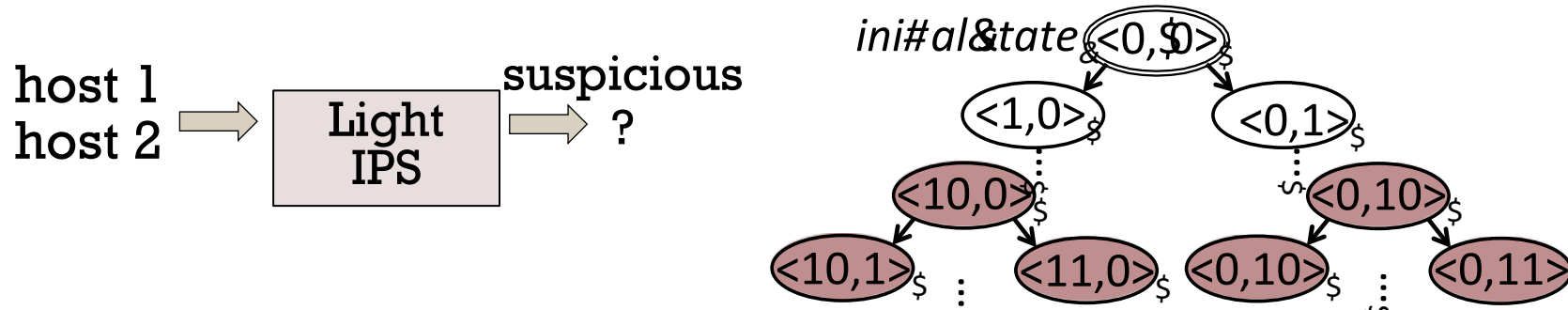




CHALLENGE 2: SCALABLE TEST TRAFFIC GENERATION

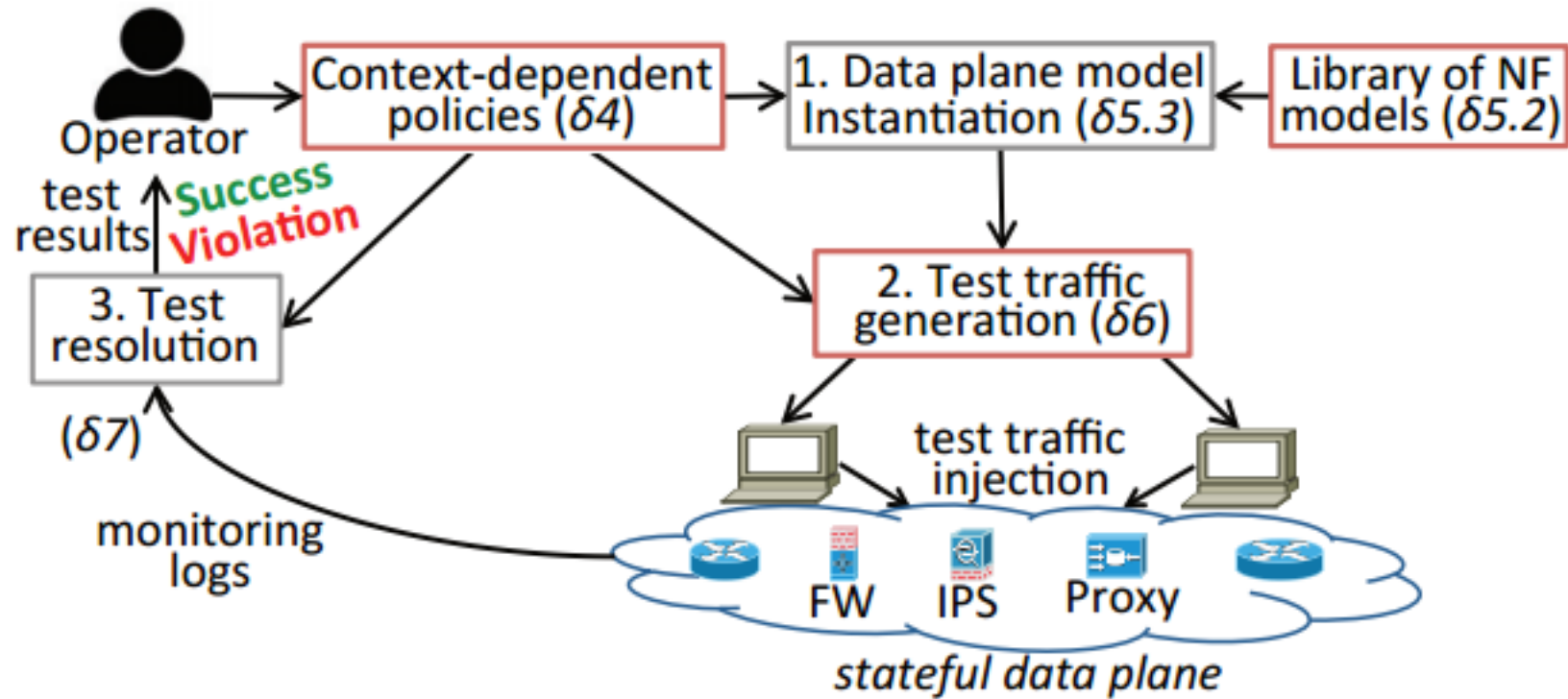


CHALLENGE 2: EXPLORING DATA PLANE STATE SPACE

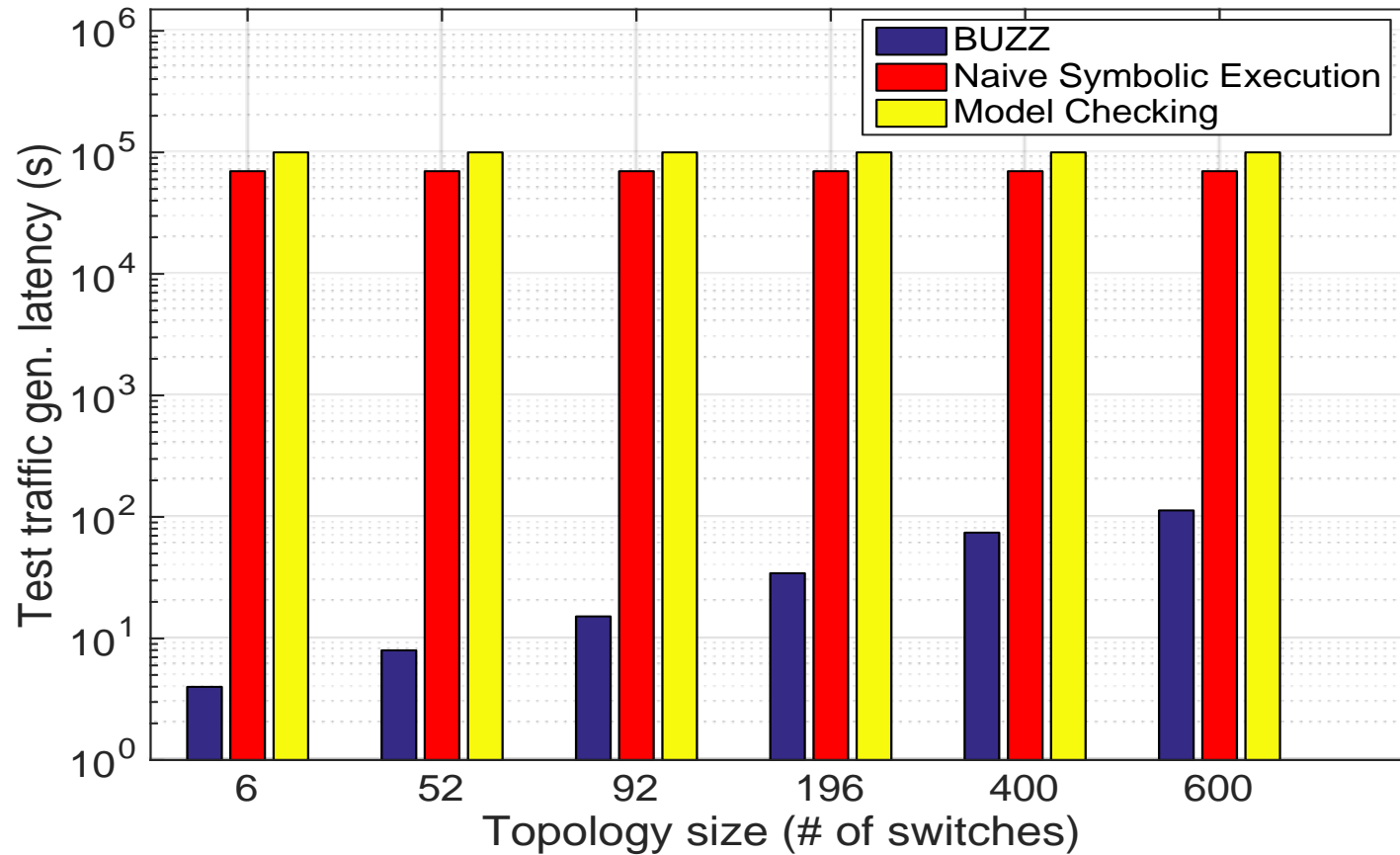


- **Conceptual view of test traffic generation:** How to reach a colored state through a sequence of traffic units?
- **Challenge of scalability** wrt traffic space and state space
 - **Strawman 1:** All possible sequences of traffic units
 - **Strawman 2:** Generate random traffic units (e.g., fuzzing)
 - **Strawman 3:** Naïve use of exploration tools (e.g., model checking)

WORKFLOW



EVALUATION: SCALABILITY OF BUZZ



Test generation takes < 2min for a network with 600 switches and 60 middleboxes

CONCLUSION

- Existing work has fundamental limitations in checking context-dependent policies in stateful data planes
- **Challenges:**
 - Expressive-yet-scalable model of stateful data planes
 - Scalable state space exploration
- **Our solution is BUZZ:**
 - BUZZ Data Unit (BDU) as traffic unit model
 - Ensemble of FSMs as a network function (NF) model
 - Scalable exploration via domain-specific optimizations
- BUZZ can help find bugs and is scalable

