# Testing Stateful and Dynamic Data Planes with FlowTest

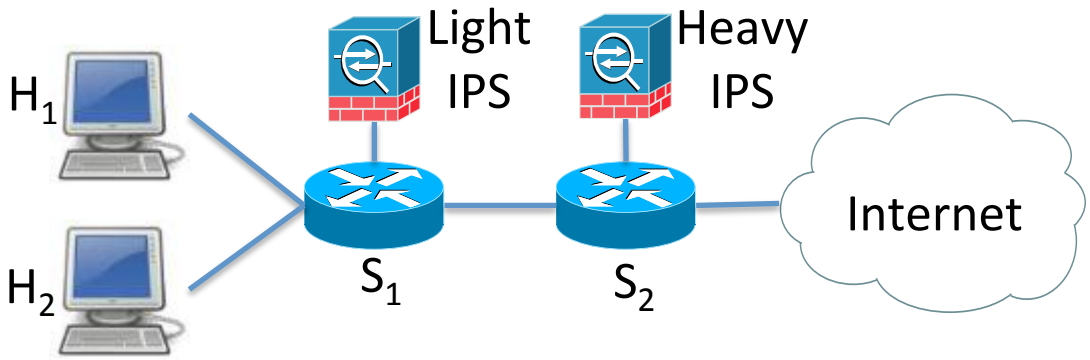
Seyed K. Fayaz, Vyas Sekar

Carnegie Mellon University

# Motivating Scenario

**Policy**

1) Keep count of TCP connections per host.
2) Deep packet inspection if a host has made too many TCP connection attempts.




How to make sure this policy is correctly implemented in the actual network?

# Existing solutions don't suffice

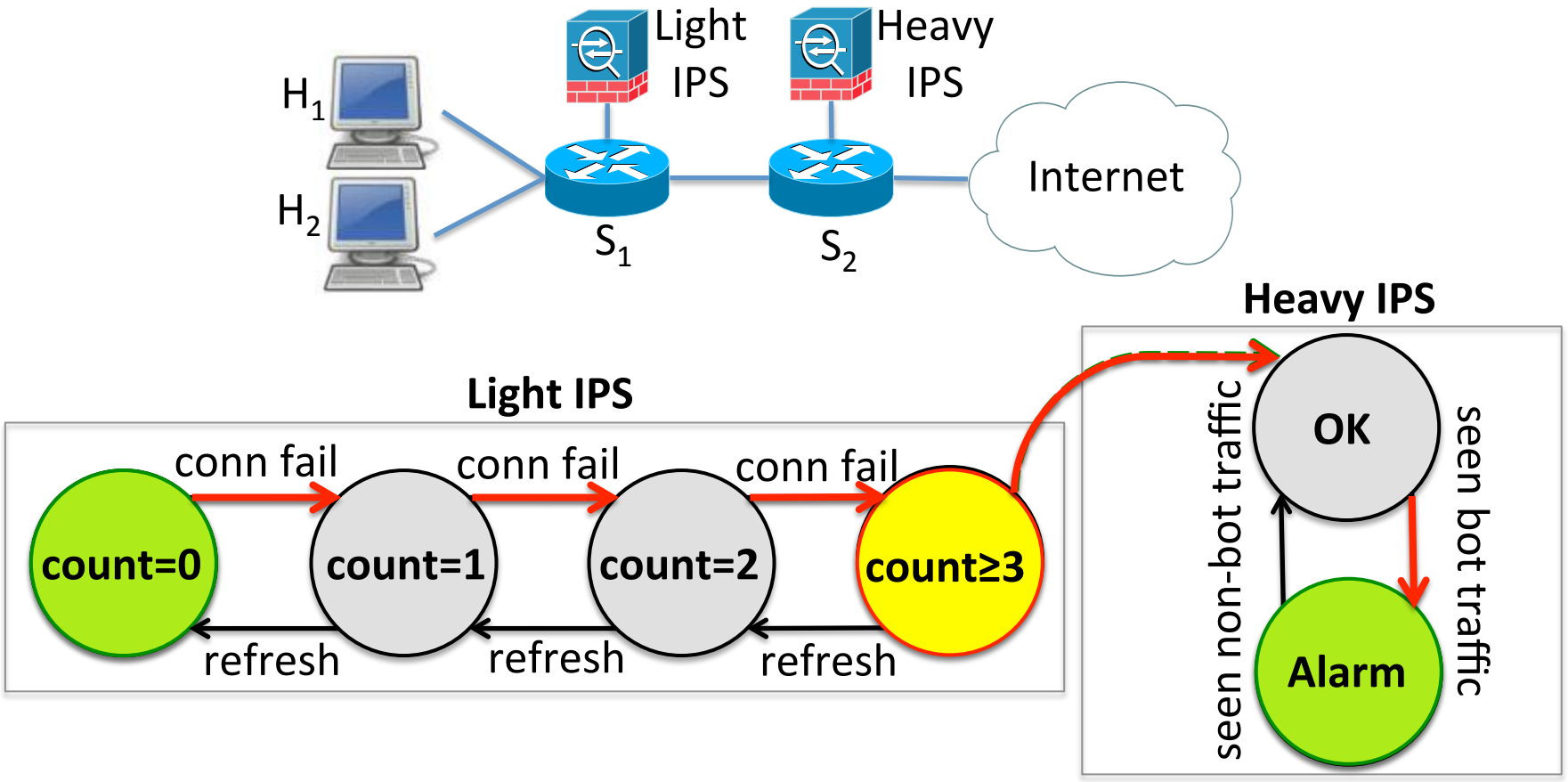
- Assume *simple, stateless* elements
  - E.g., switches and simple ACL devices
- Work with *static and context-free* policies
  - E.g., reachability
  - E.g., access control
- Focus on *single packet* effects

# Our Approach: FlowTest

- FlowTest's approach: *testing* the data plane

- We need:
  - A *model of the entire data plane*
    - Including middleboxes
  - To generate *test scenarios* that exercise
    - Data plane states
    - Policy contexts
  - To *monitor* and *validate* test results

# Early Promise




Generating test traffic can be formulated using *AI planning*.

We validated our solution using an SDN prototype.

# Conclusions

- Real world networks are complex
  - Stateful elements
  - Dynamic and contextual policies
- We argue for testing data planes that incorporates data plane models
- Initial promise of FlowTest via FSMs and planning
- Many open challenges