SNAP: Stateful Network-Wide Abstractions for Packet Processing

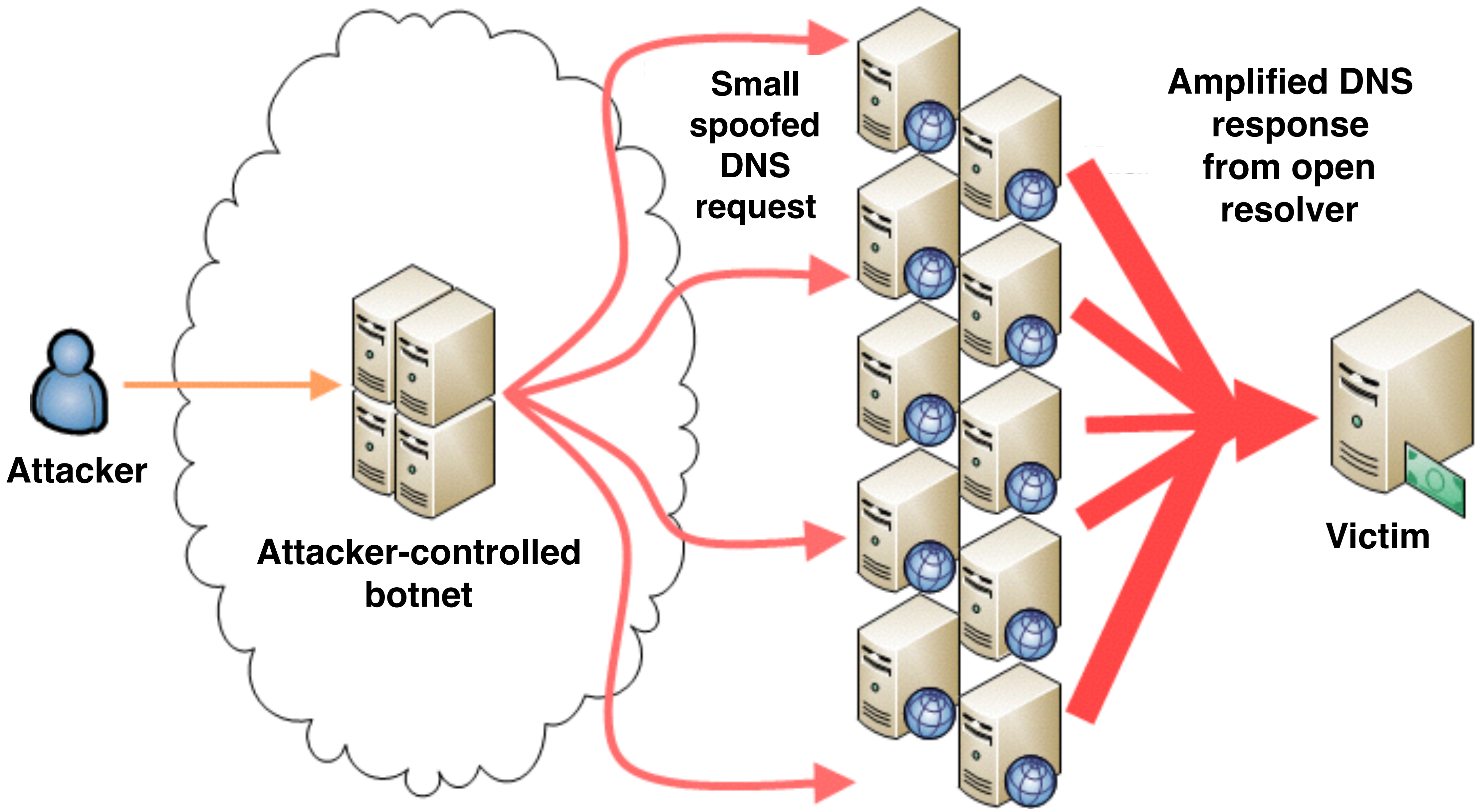
Mina Tahmasbi Arashloo[1], Yaron Koral[1], Michael Greenberg[2], Jennifer Rexford[1], and David Walker[1]

[1] Princeton University, [2] Pomona College

Two-tiered Programming Model

1. **Stateless** OpenFlow rules
    - process each packet independently
2. **Stateful** component
    - on top of controller
    - changes rules based on current state

Example - Detecting DNS Reflection Attacks



http://ddosandbotnets.blogspot.com

Example - Detecting DNS Reflection Attacks [1]

- For each host
  - Record DNS **requests**
  - Increment a counter for **unmatched responses**
  - Mark as **suspicious** after a threshold

[1] Bohatei: flexible and elastic DDoS defense, Fayaz et.al.

Practical Concerns

- Cannot send every packet to the controller
- Cannot have **per-packet stateful** processing
    - decide what to do with the packet based on packets seen so far!

Opportunity:
Local State on Data Plane

- Programmatic control over local state
    - P4, OpenState, POF, Open vSwitch
- Simple stateful network functions can be offloaded to programmable switches!

Opportunity:
Local State on Data Plane

- Programmatic control over local state
  - P4,
- Simpl loaded to programmable switches!

Too low level!

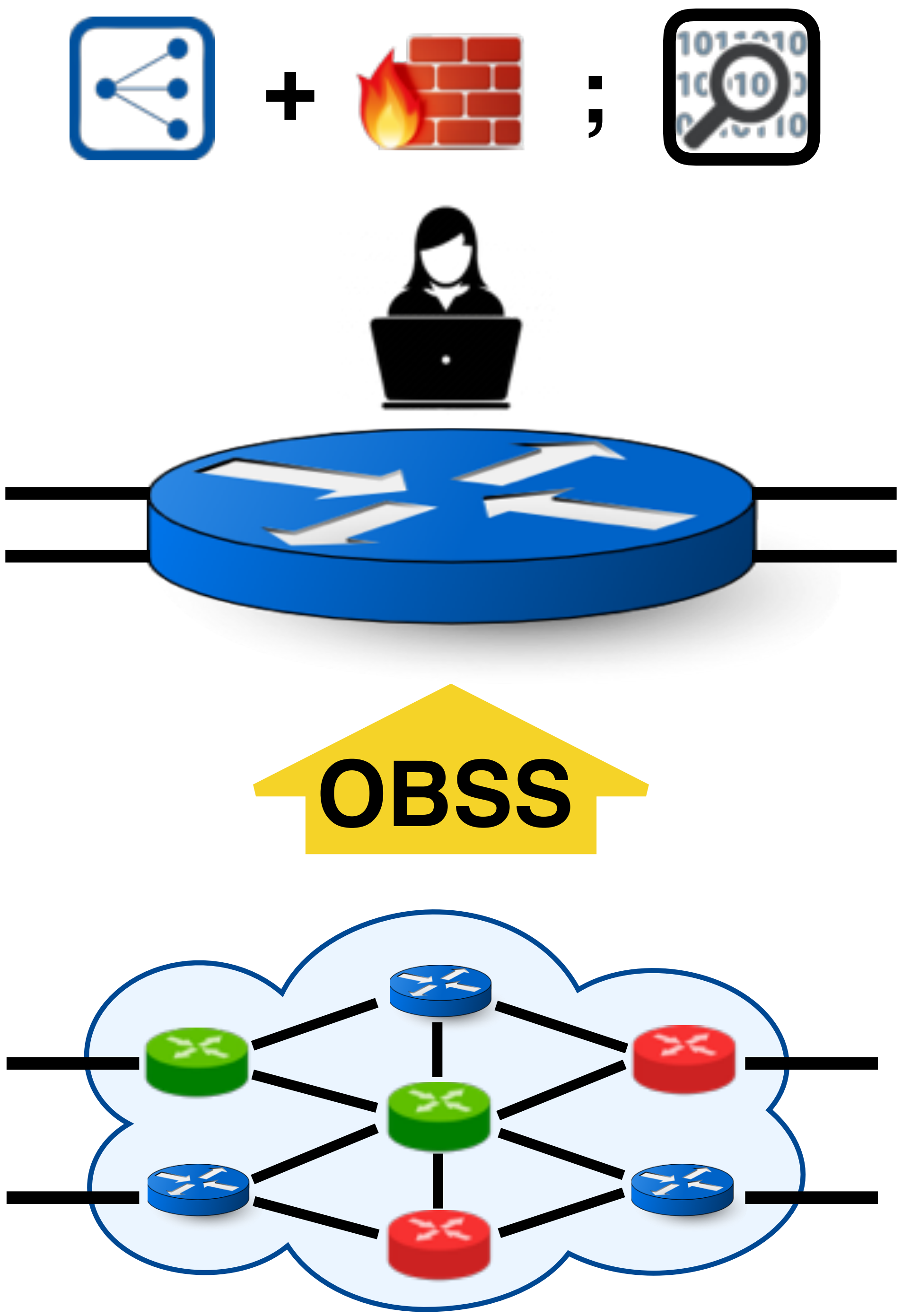
Our Goal

1. **High-level programming language**
    - one big **stateful** switch!
2. **Compiler**
    - to automatically distribute stateful programs

Our Contribution

SNAP Language and Compiler

SNAP - Language



- One Big Stateful Switch **(OBSS)**
- Composition

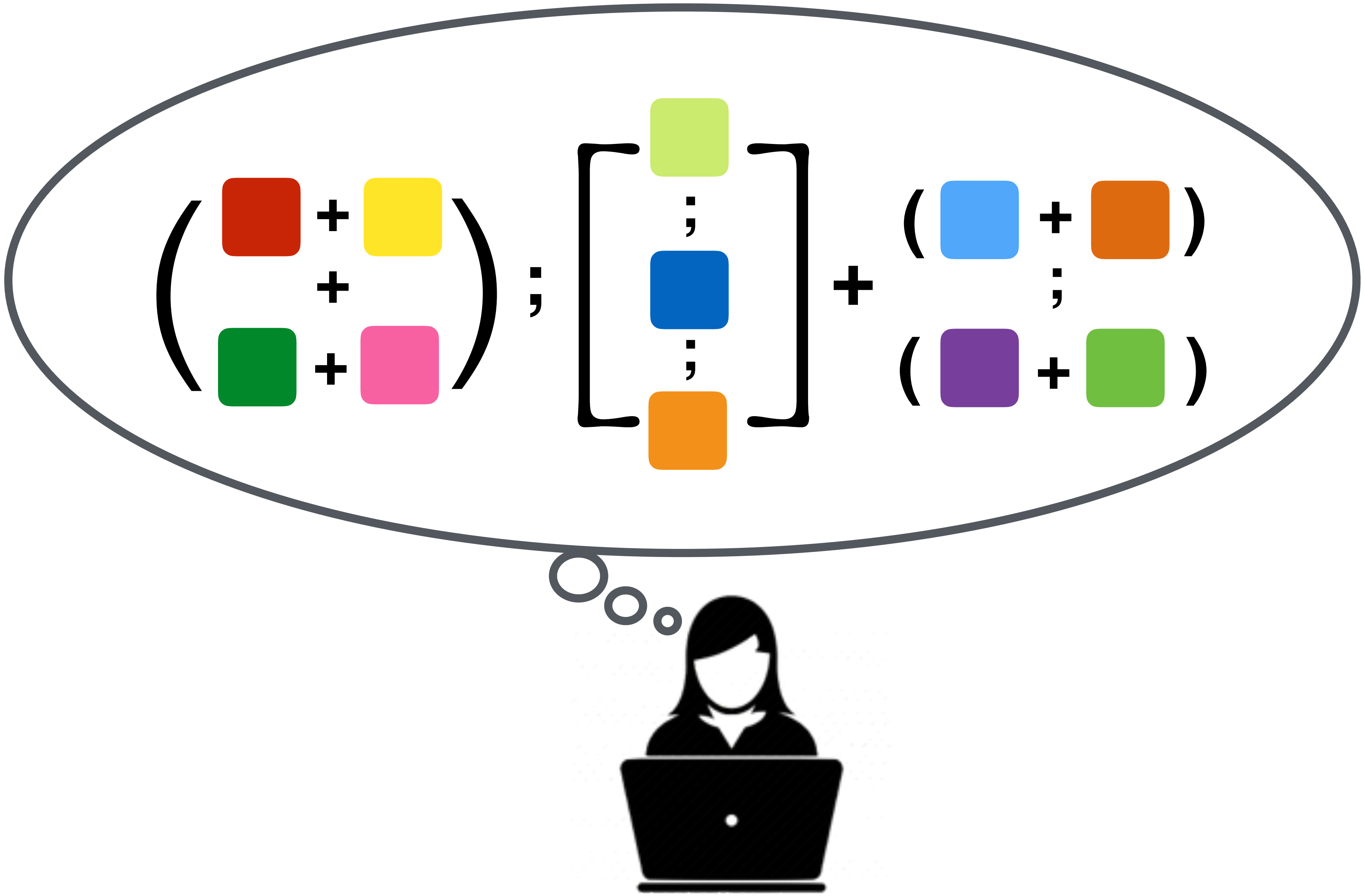
DNS Reflection Detection in SNAP

```
if srcip in CSNET & dstport = 53 then
   seen[srcip][dns.id] ← True
else if dstip in CSNET & srcport = 53 then
   if ~seen[dstip][dns.id] then
      unmatched[dstip]++;
      if unmatched[dstip] = threshold then
         susp[dstip] ← True
   else id
else id
```

DNS Reflection Detection in SNAP

```
if srcip in CSNET & dstport = 53 then
   seen[srcip][dns.id] ← True
else if dstip in CSNET & srcport = 53 then
   if ~seen[dstip][dns.id] then
      unmatched[dstip]++;
      if unmatched[dstip] = threshold then
         susp[dstip] ← True
   else id
else id
```

Composition



SNAP - Compiler

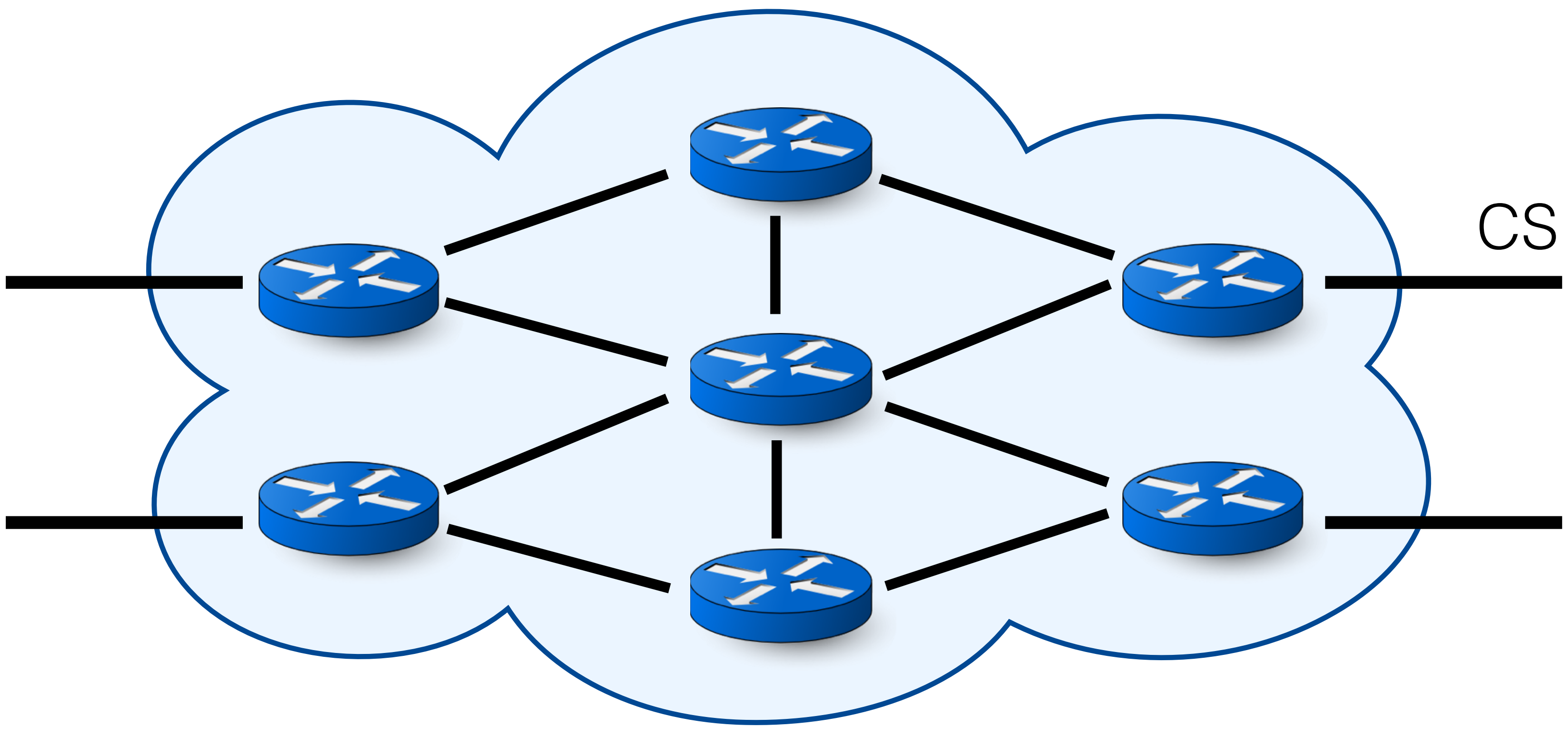
where to place state variables

how to forward packets through them

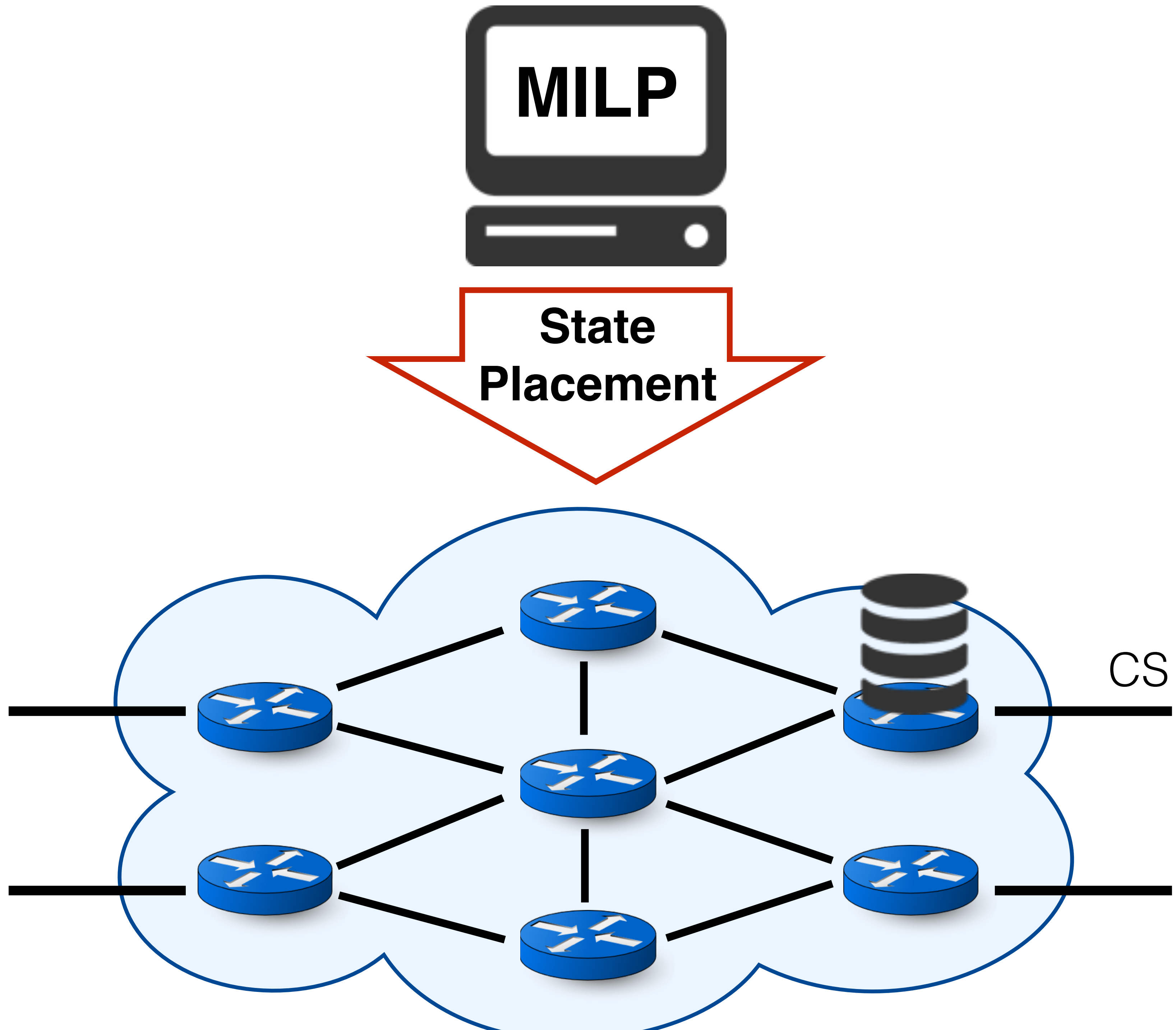
Program Analysis

- Each flow should go through switches with
    - all the state variables that it needs
    - in the correct order

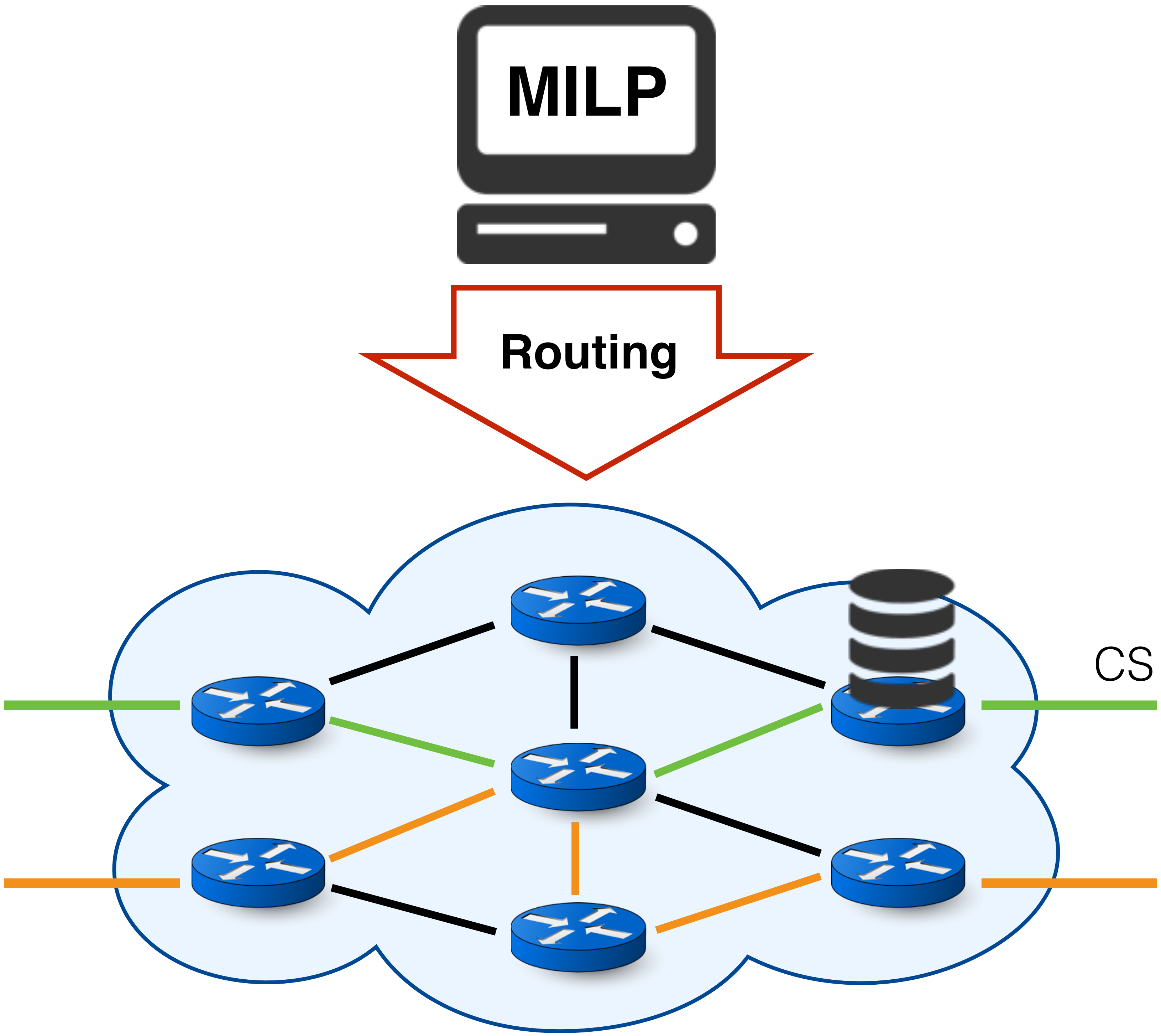
Mixed-Integer Linear Program (MILP)



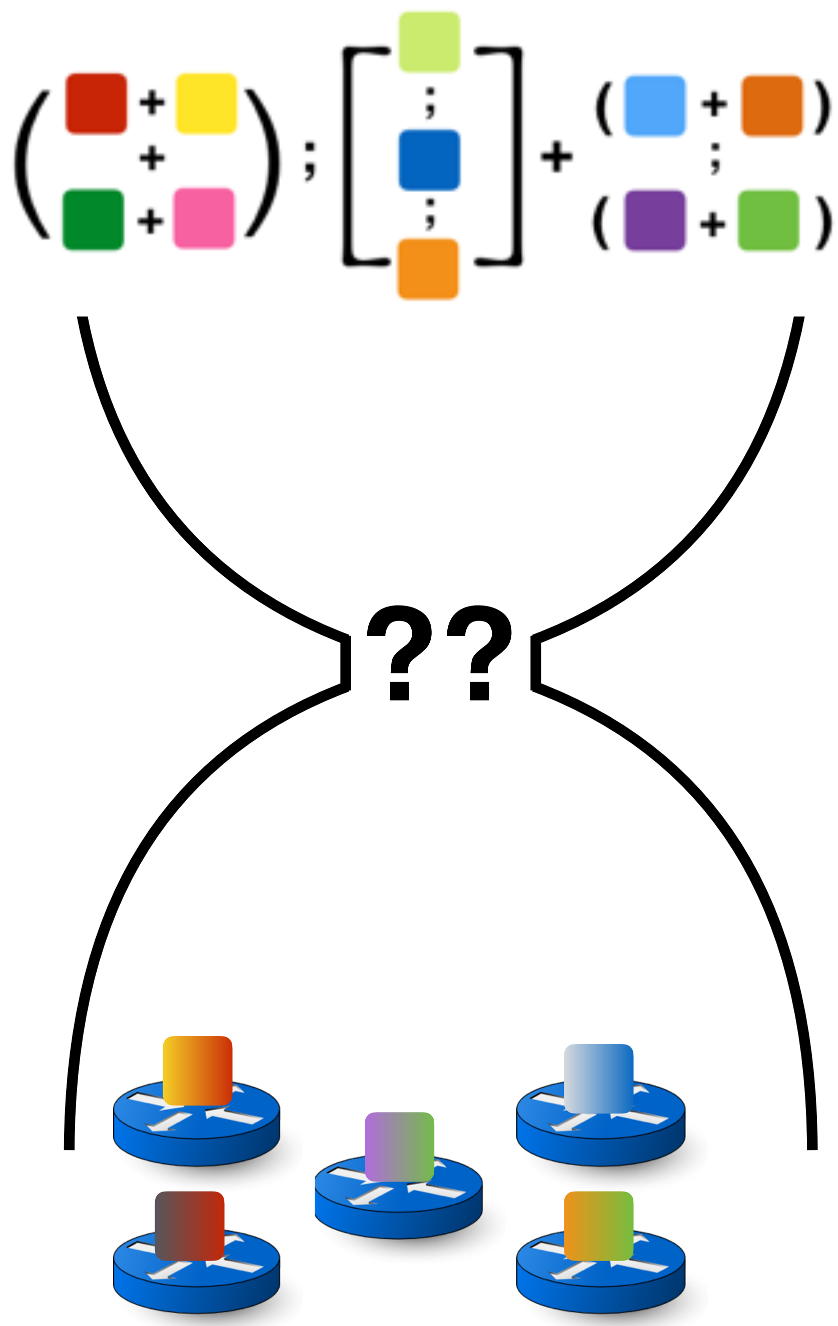
Mixed-Integer Linear Program (MILP)



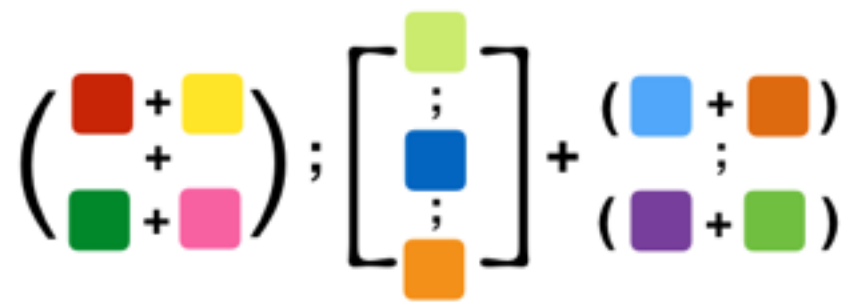
Mixed-Integer Linear Program (MILP)



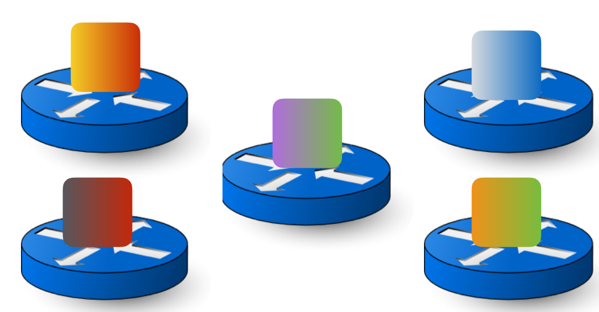
Intermediate Representation (IR)



Intermediate Representation (IR)



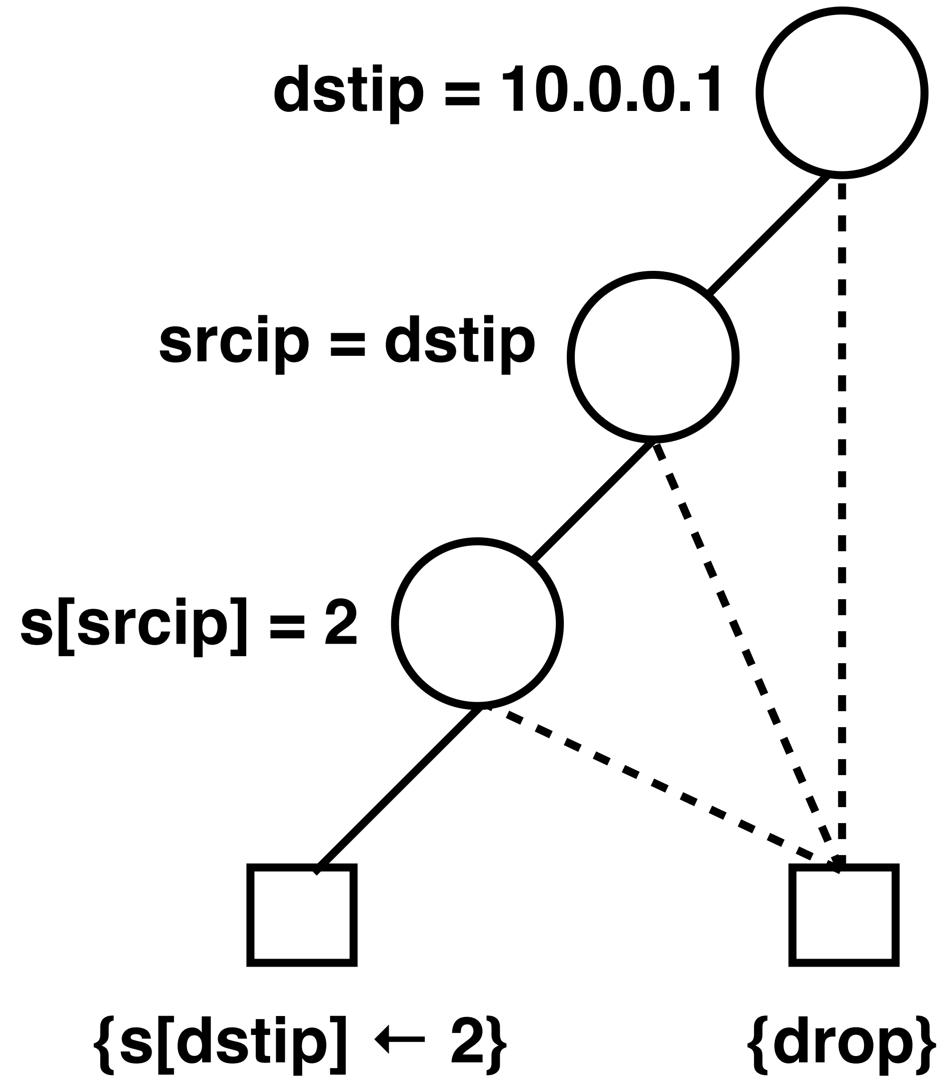
Composable and Easily Partitioned IR



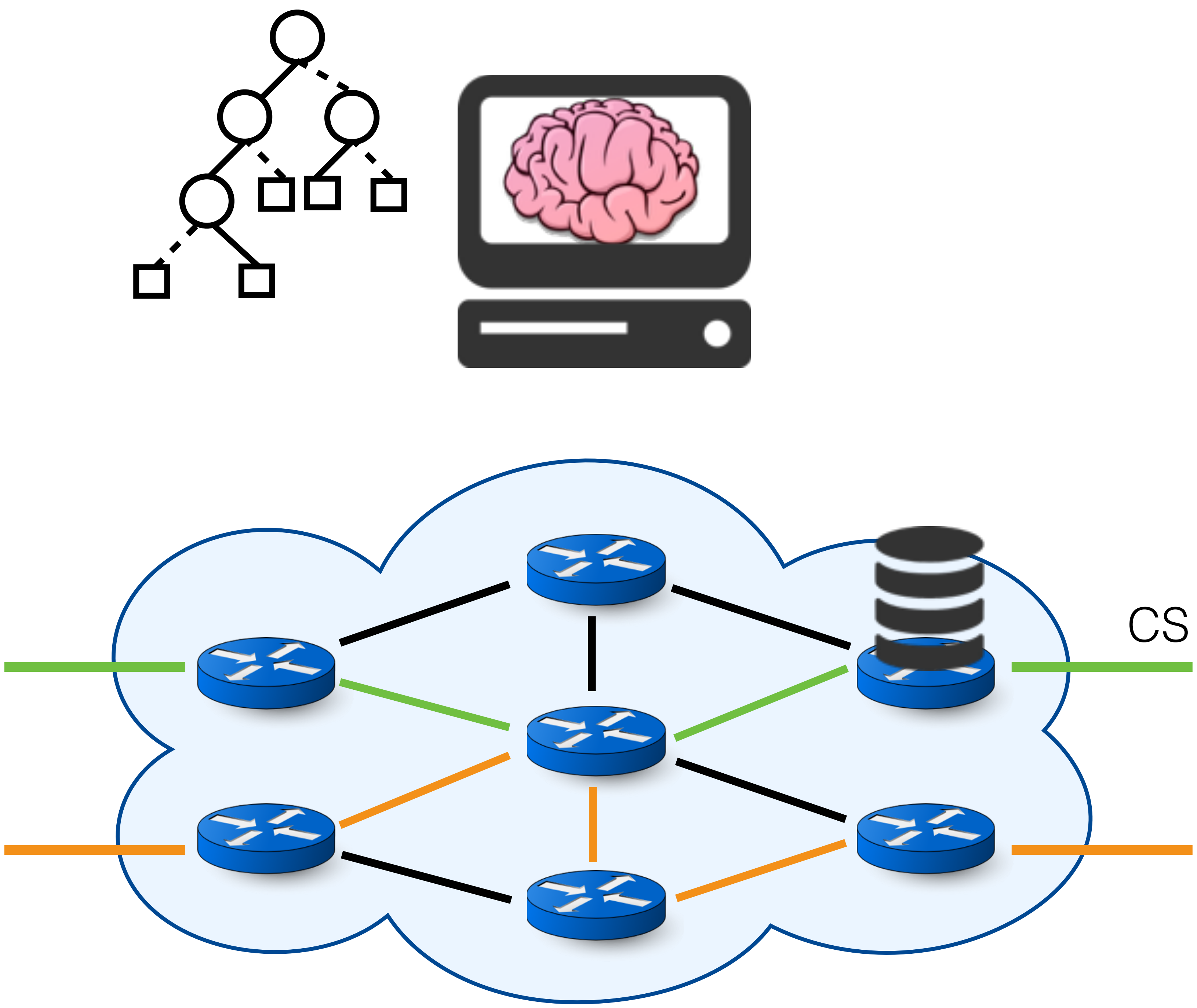
Extended Forwarding Decision Diagrams (xFDDs)

- Intermediate node:
  test on header fields and state

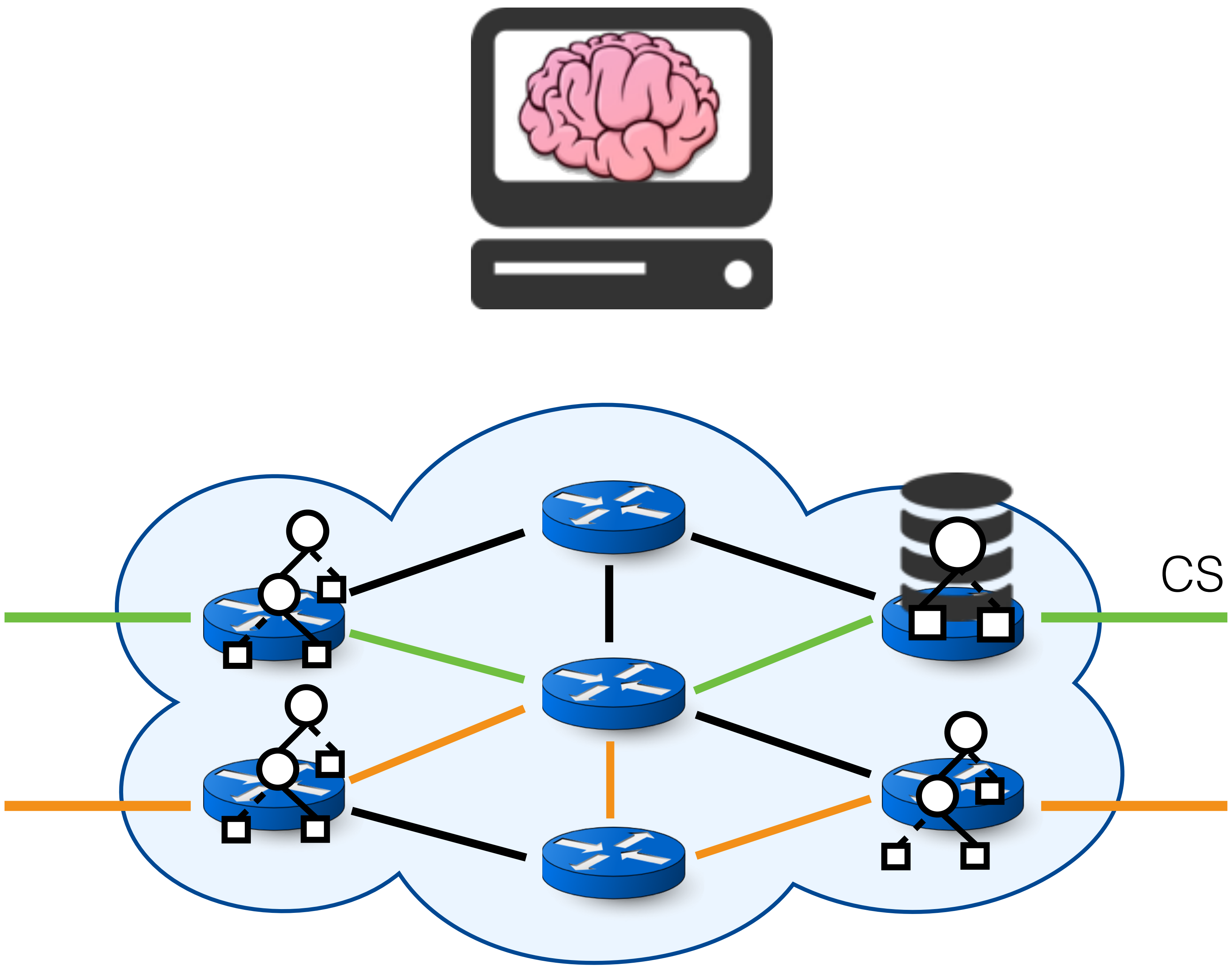
- Leaf: set of action sequences



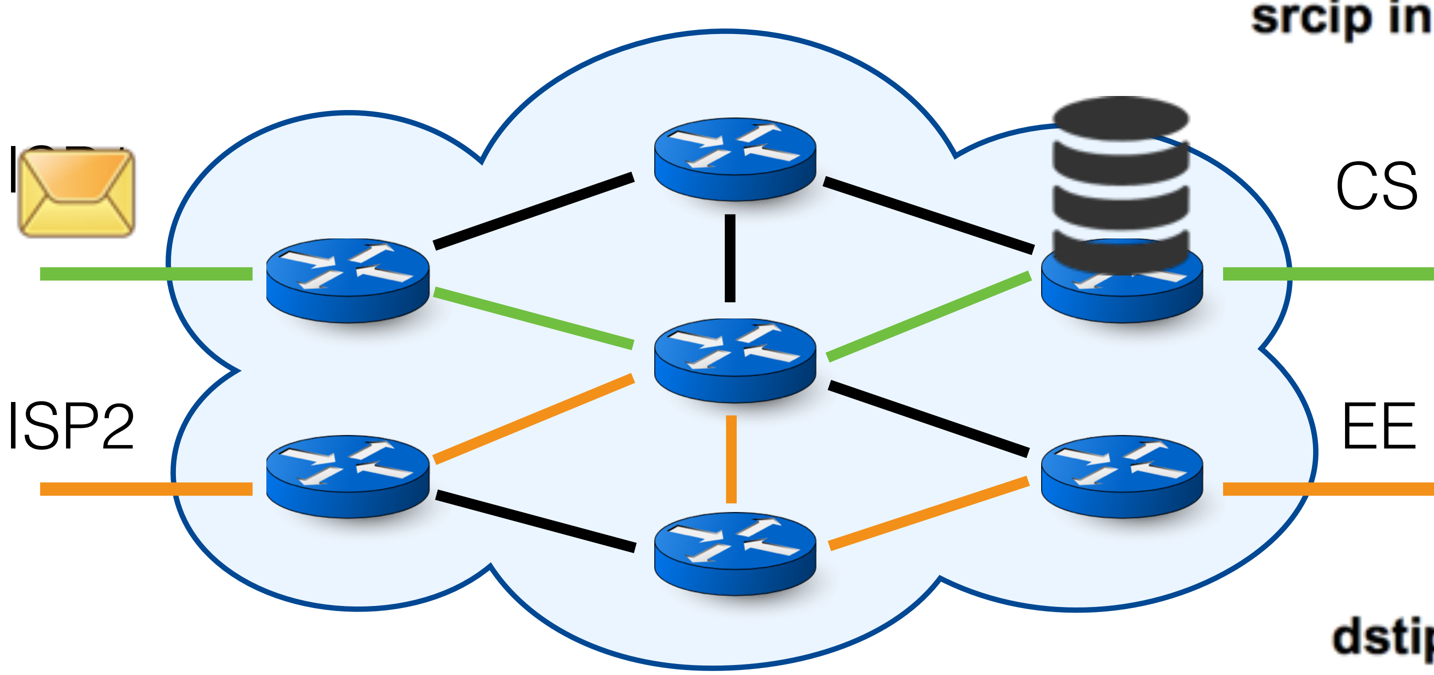
Distributing the xFDD

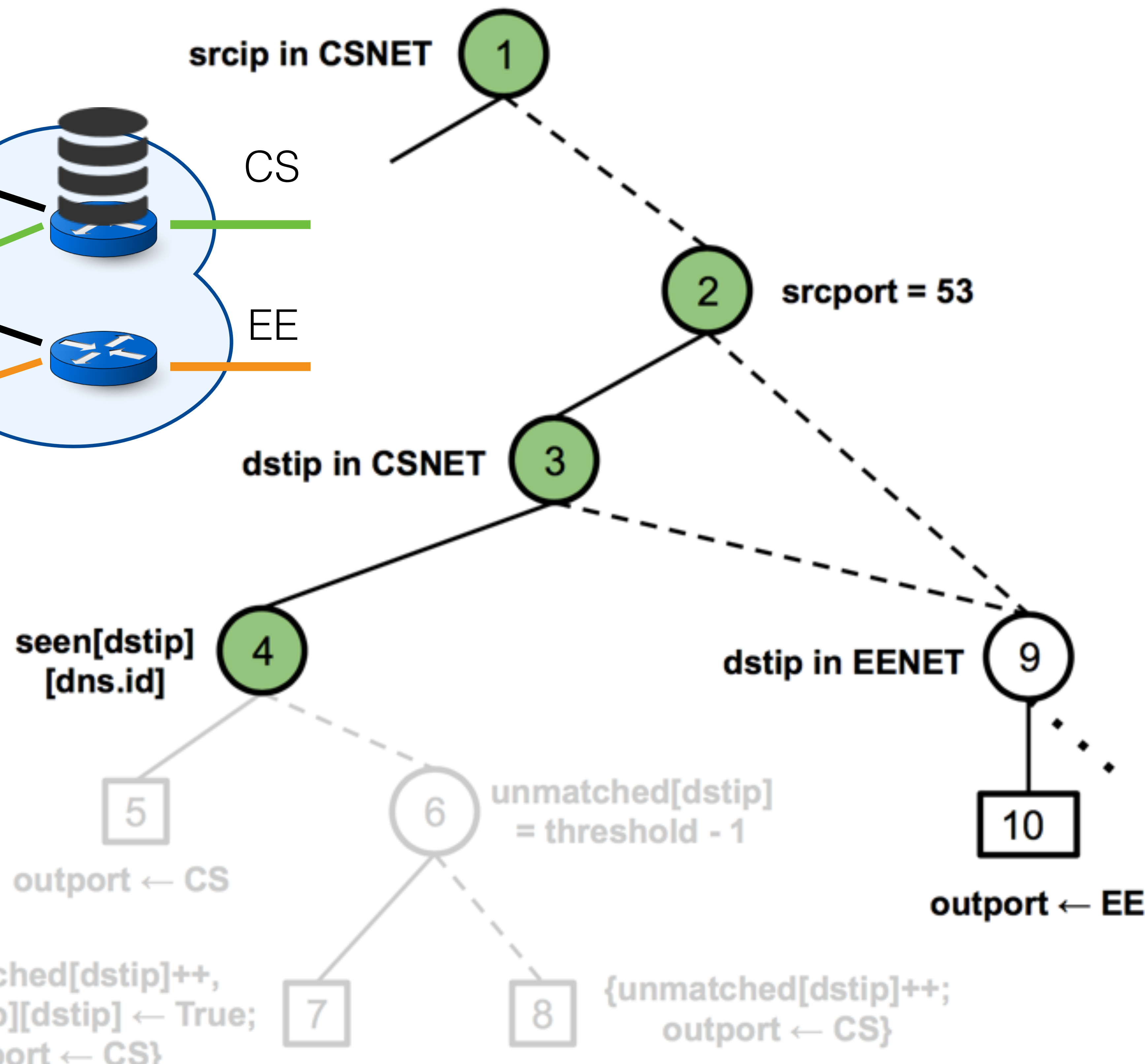


Distributing the xFDD

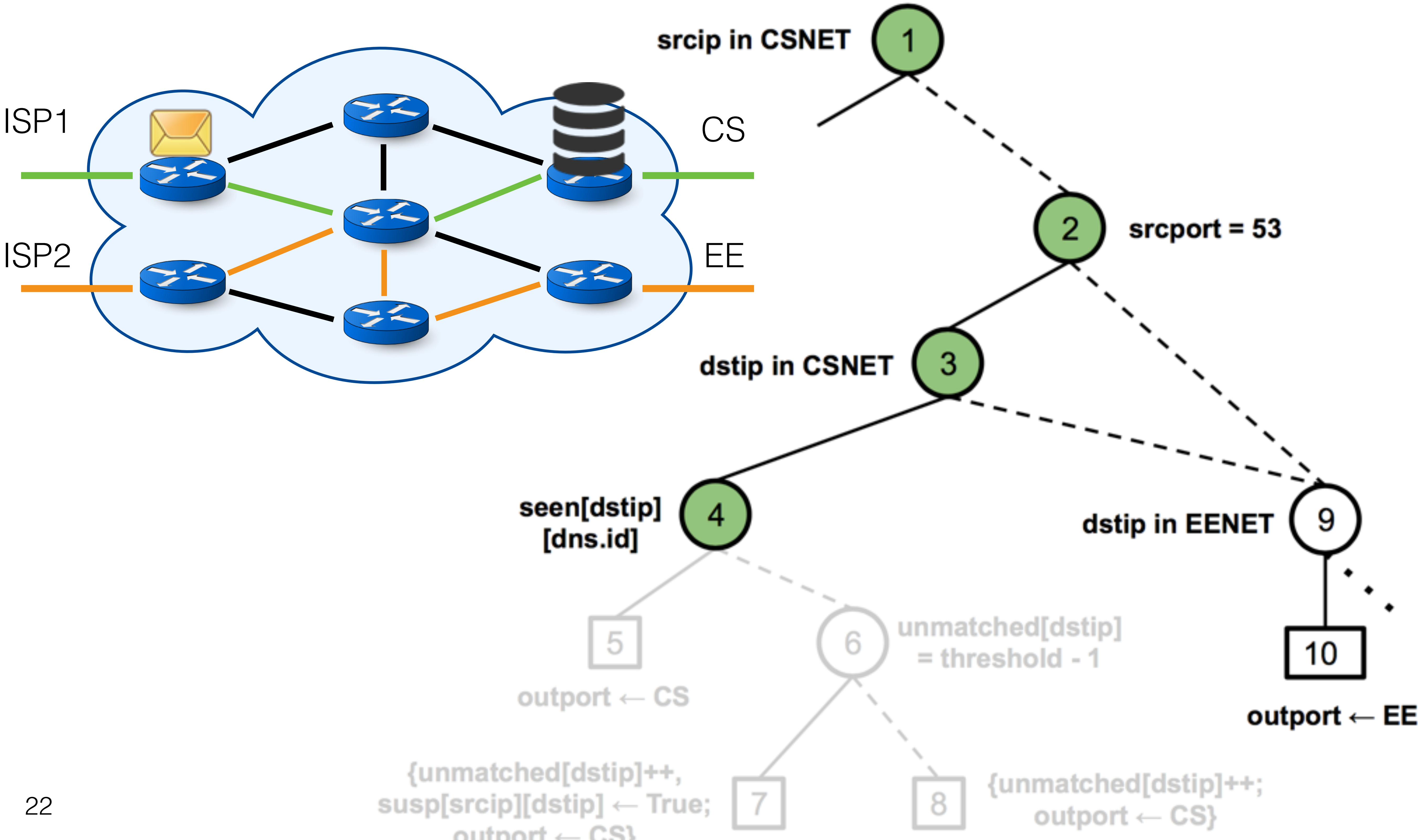


Putting It All Together

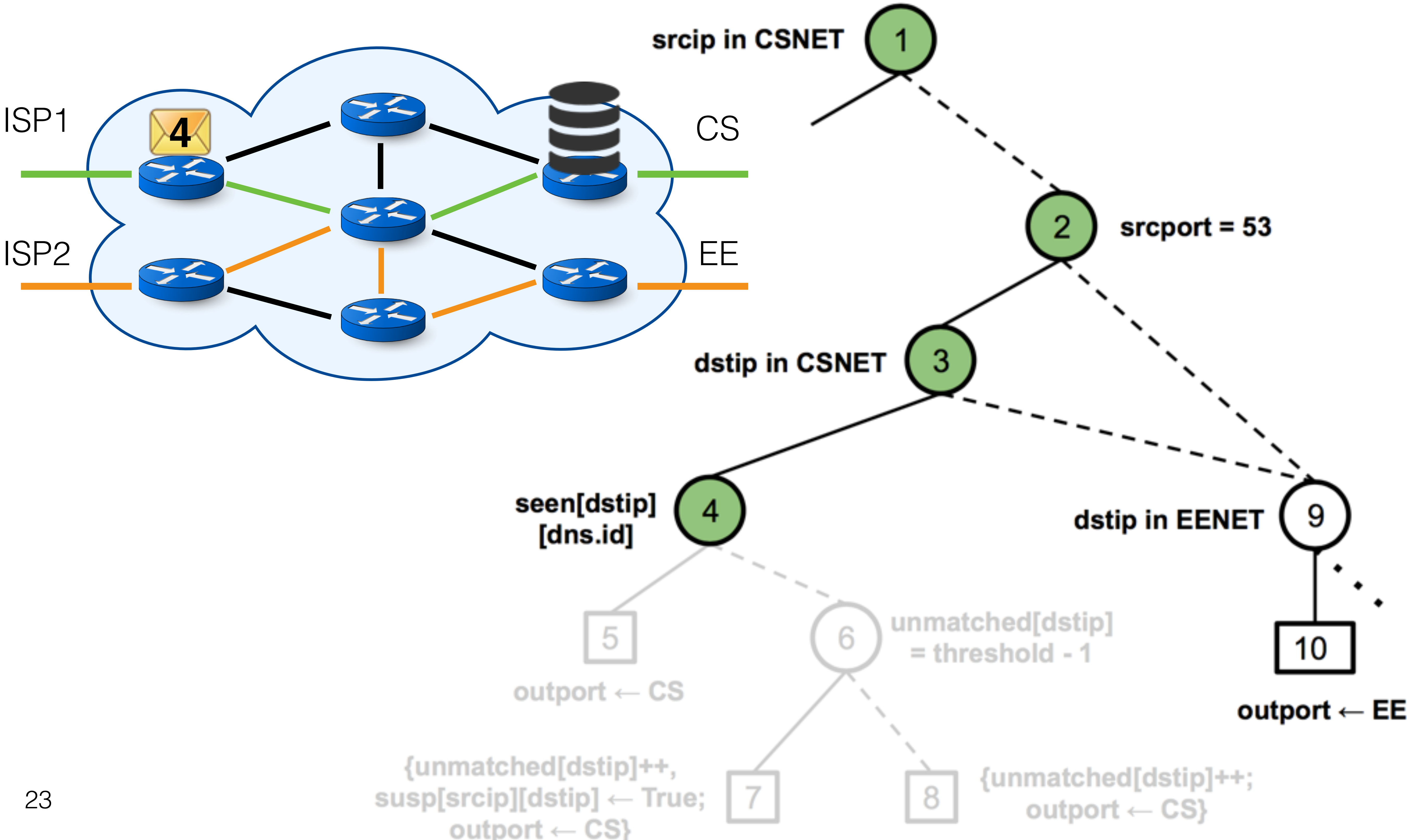




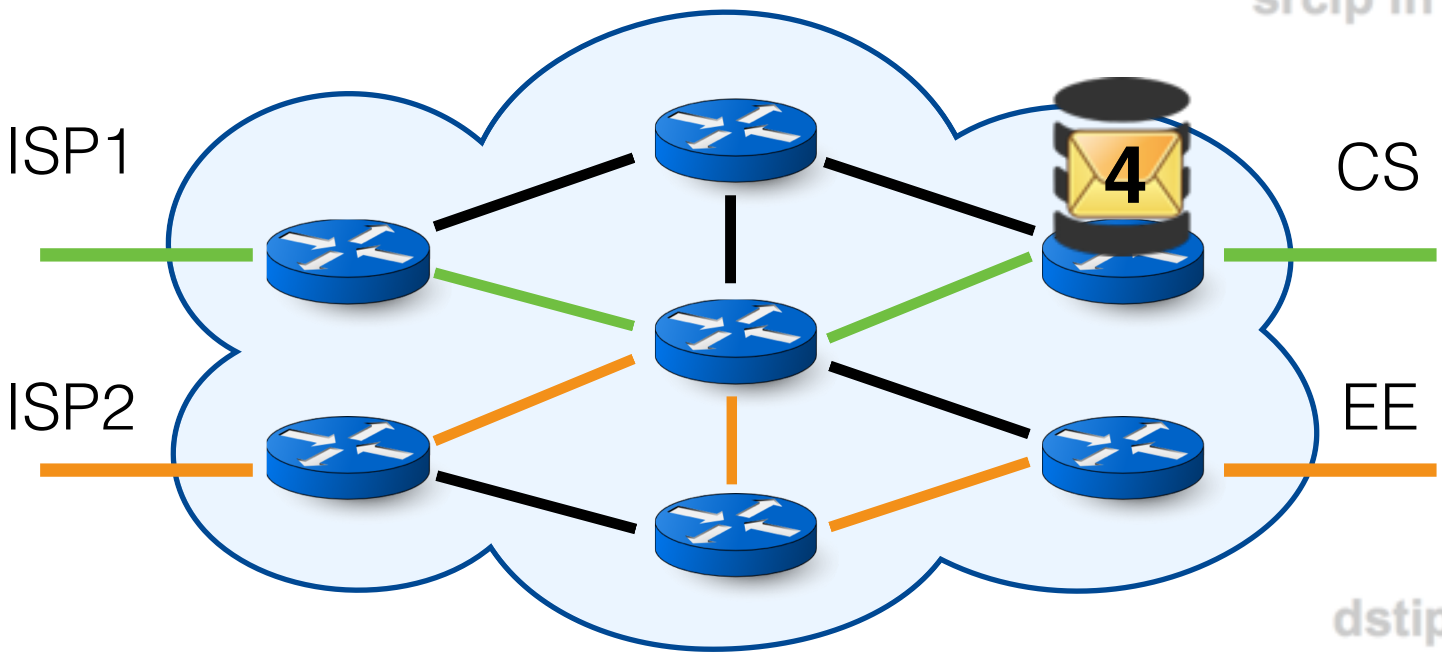
Putting It All Together

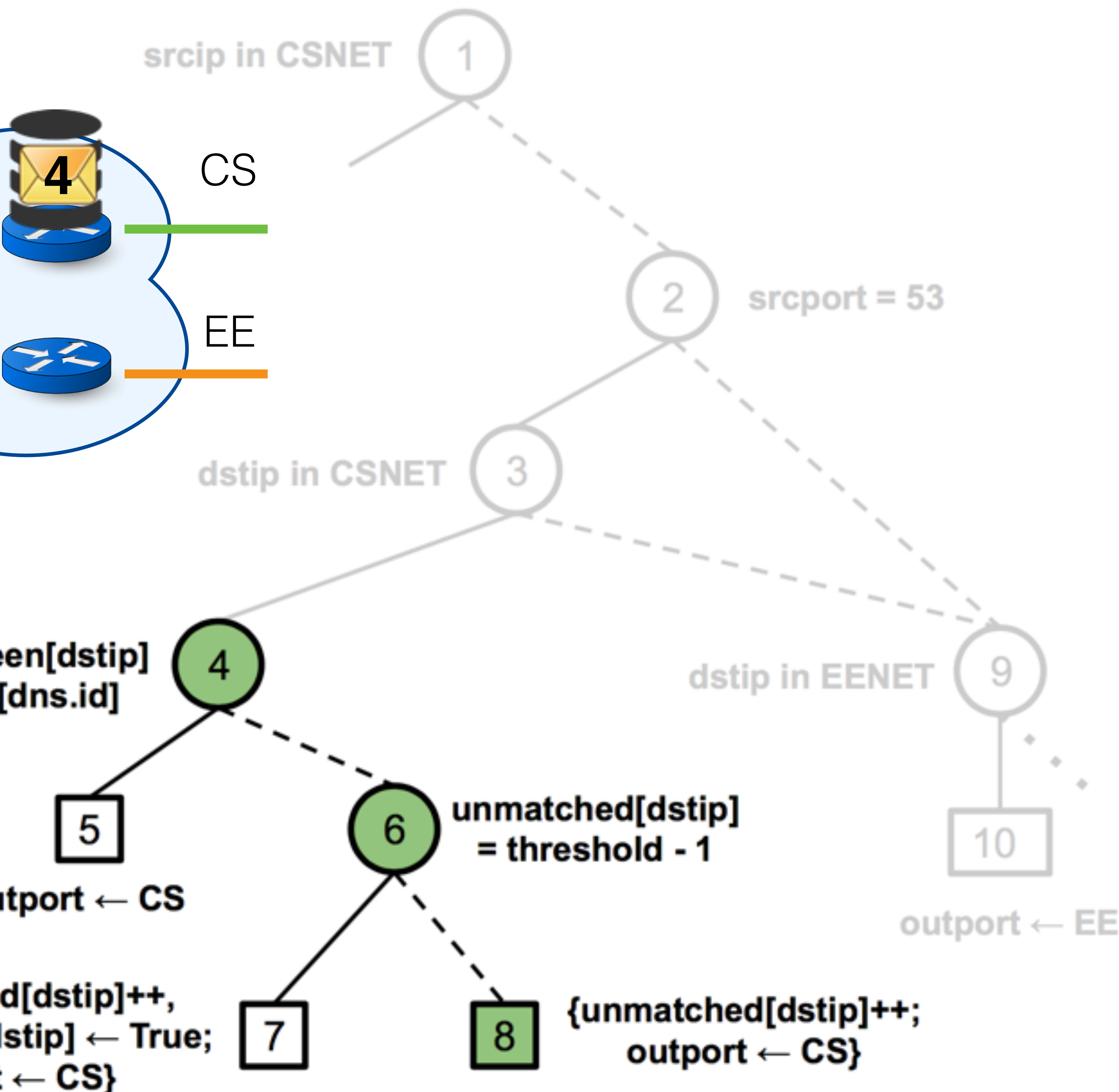


Putting It All Together

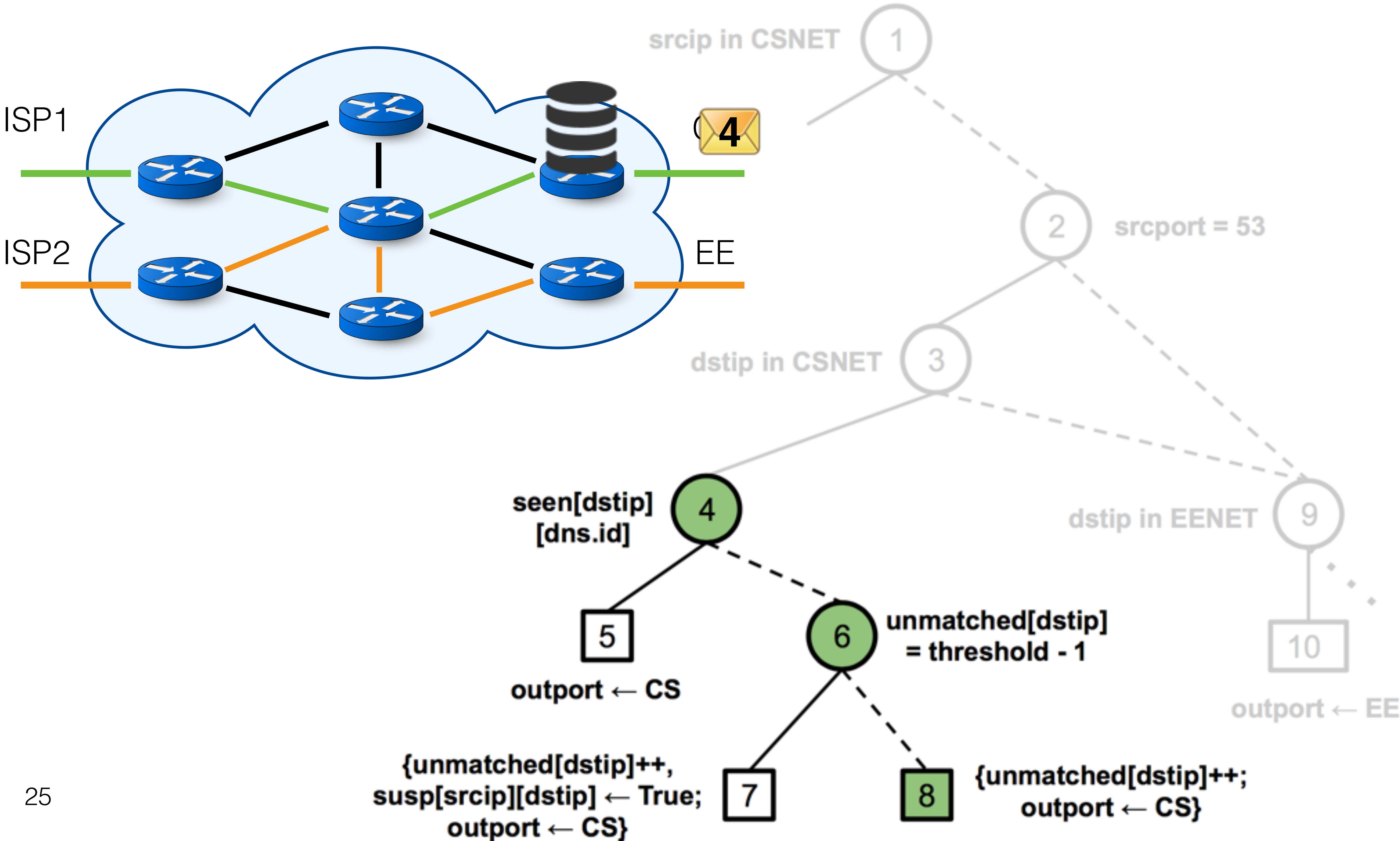


Putting It All Together





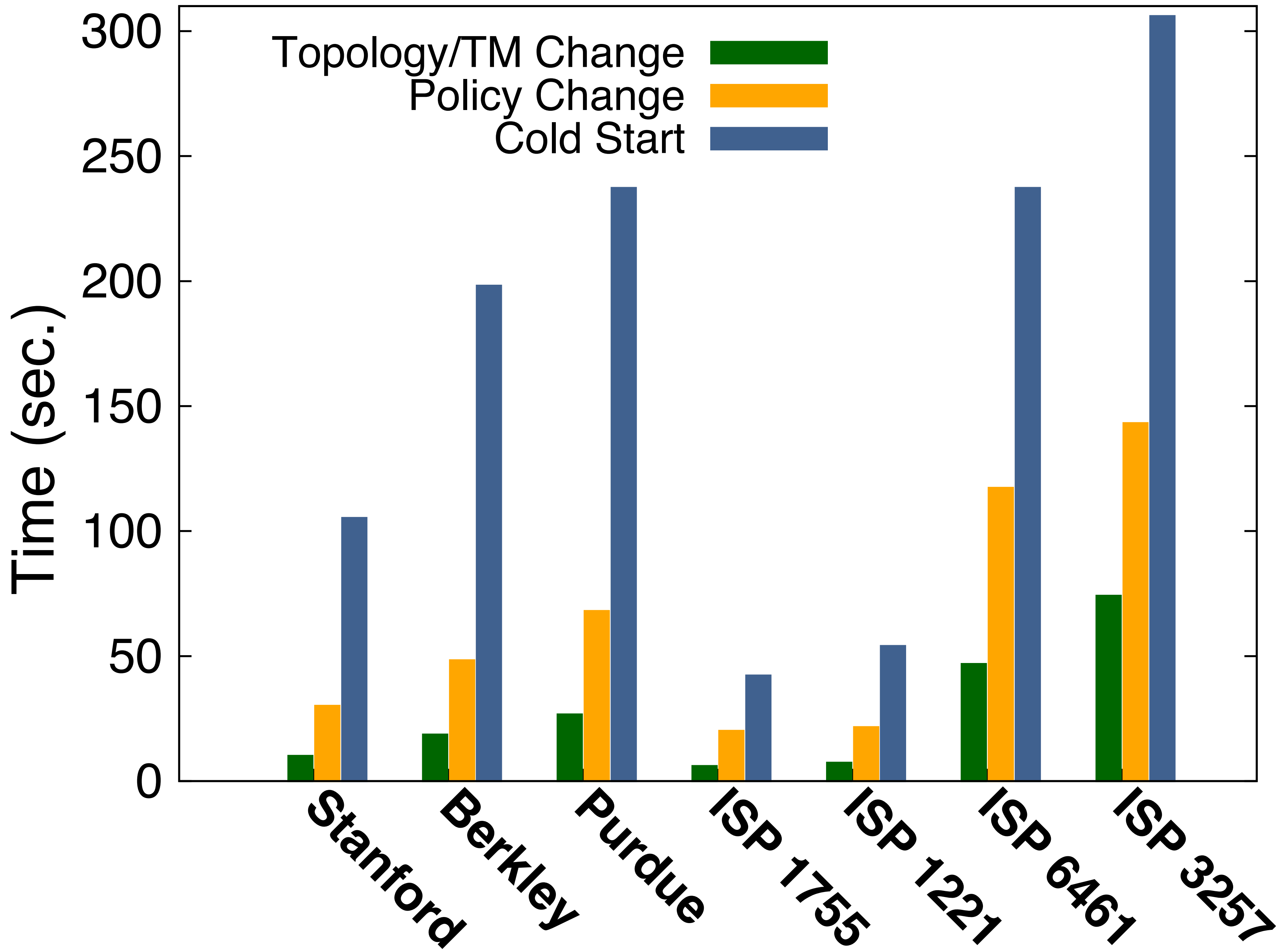
Putting It All Together



Compiler Evaluation - Benchmarks and Scenarios

- Topologies with 100s of switches and edges
  - 7 campus and ISPs
- Scenarios
  - Cold start
  - Policy change
  - Topology/TM change

Compiler Evaluation - Results



Thank You!

Questions?