A COMPRESSIVE METHOD FOR MAINTAINING FORWARDING STATES IN SDN CONTROLLER

Ying Zhang*, Sriram Natarajan$, Xin Huang^, Neda Beheshti*, Ravi Manghirmalani*
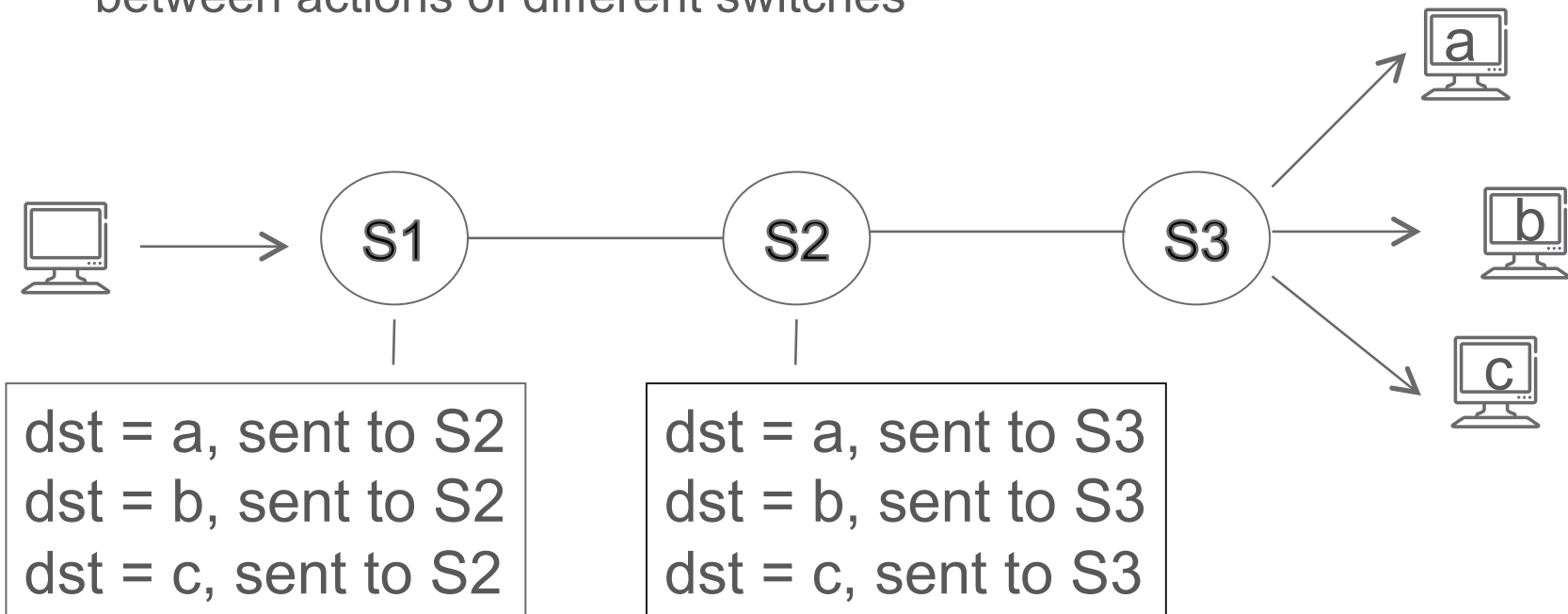
Ericsson Research*, NTT$, CYAN^

Controller needs to maintain a copy of forwarding states

› Forwarding states in SDN: tables on the switches

› Most existing controllers do not maintain a copy of forwarding states

› There is a need in real-world deployment scenario

  – Fast fault recovery from transient switch failures

  – Network state queries by multiple control applications

  – Consistency check on the switches' actual states

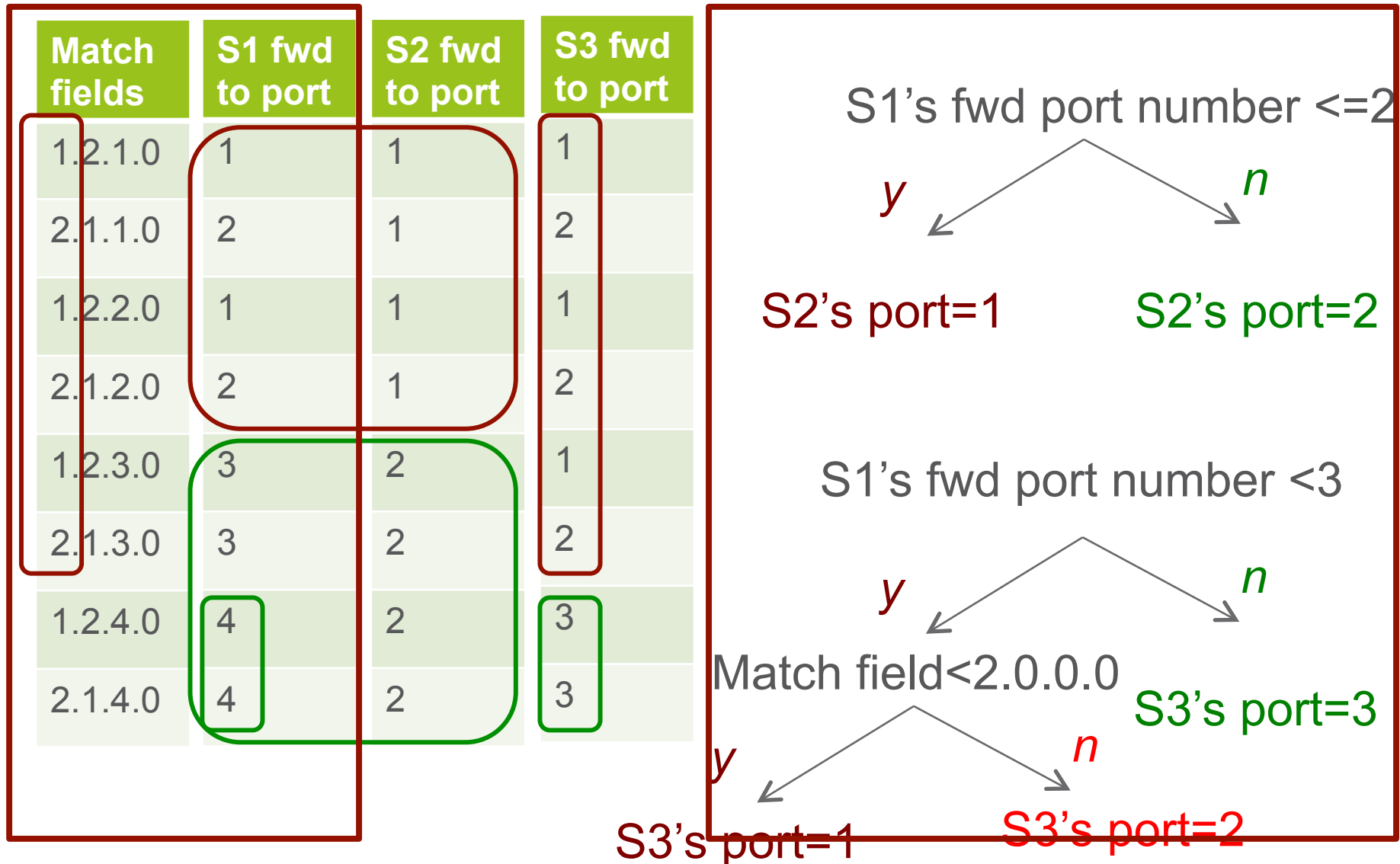 – Rule space analysis for optimization, debugging, etc.

Need an efficient solution to store states in controller

› Naïvely storing all the tables from all switches takes huge space

› Explore correlation between switches
  - Intuition: because of topological properties, there exists dependency between actions of different switches

Use prediction tree to capture dependency

Match fields	S1 fwd to port	S2 fwd to port	S3 fwd to port
1.2.1.0	1	1	1
2.1.1.0	2	1	2
1.2.2.0	1	1	1
2.1.2.0	2	1	2
1.2.3.0	3	2	1
2.1.3.0	3	2	2
1.2.4.0	4	2	3
2.1.4.0	4	2	3

S1's fwd port number <=2

y   n

S2's port=1   S2's port=2

S1's fwd port number <3

y   n

Match field<2.0.0.0   S3's port=3

y   n

S3's port=1   S3's port=2

System design and results

Table combination	Dependency discovery	Compression
- Handling empty entries - Handling multiple tables	- Use Bayesian network to identify dependencies between columns - Build classification and regression tree (CaRT)	- Divide columns to predicting and predicted sets - Support incremental update

› Evaluation on Mininet with NOX

› Using Bayesian network and C4.5 decision tree for dependency discovery

› Results:

   − Regular topology: 1.77% of original size

   − Realistic topology: 2.8% of original size

Q&A?