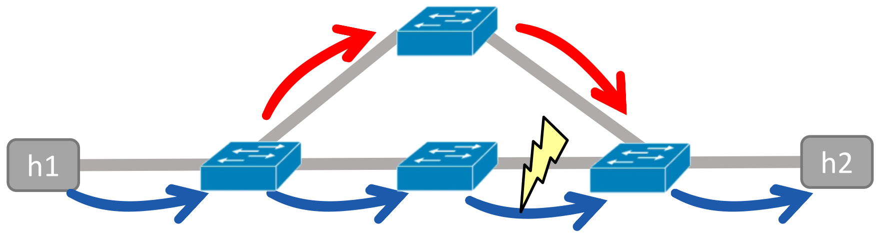
FAILURE RESILIENCY WITH STATEFUL SDN DATA PLANE

- Forwarding pipeline design to allow
  - E2E proactive protection independent from controller reachability
  - Fully programmable failure detection and recovery in the fast-path
- Programmable failure detection
  - BFD-like
- Fast reroute
  - Inspired by MPLS
  - for both local and non-local preplanned failures
- Based on stateful data plane
  - OpenState
  - OVS + learn action
  - $P4_{14}$



POLITECNICO MILANO 1863

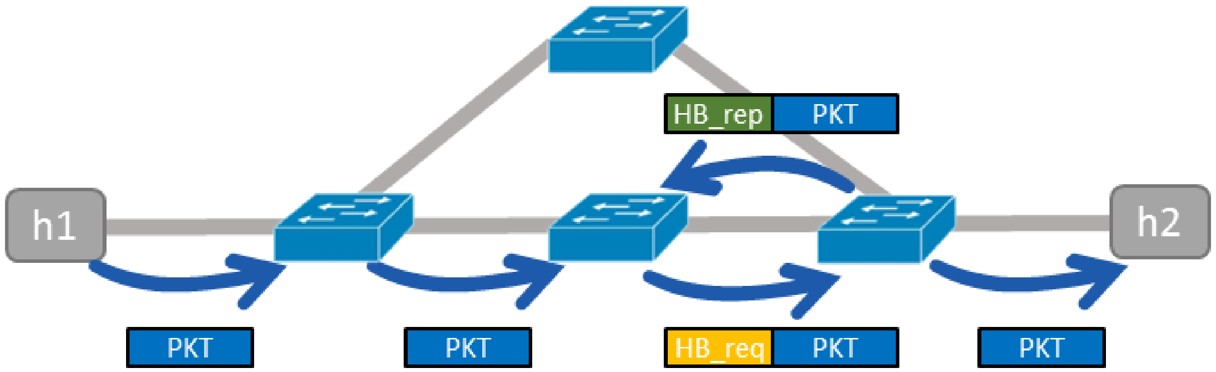
Beba BEhavioural BAsed forwarding

C. Cascone, L. Pollini, D. Sanvito, A. Capone, B. Sansò - *SPIDER: Fault Resilient SDN Pipeline with Recovery Delay Guarantees* - IEEE NetSoft 2016

Failure detection

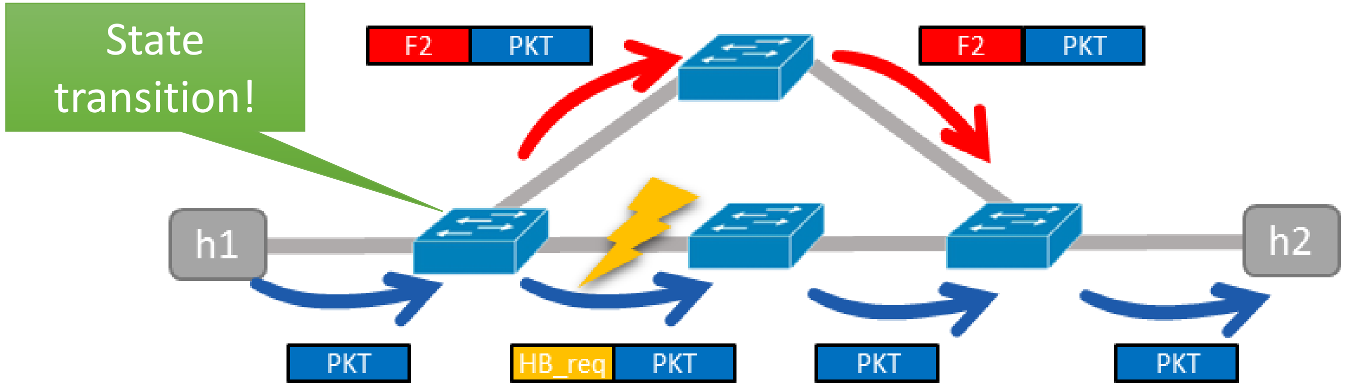
Assumption:

As long as packets are received from a given port, that port can be also used to transmit packets



- If no packet is received from port x within a δ_1 interval:
  - Next data packet towards port x is tagged with a special value (Heartbeat request)
  - Port x is declared down if adjacent node does not send back a copy (Heartbeat reply) within a δ_2 interval
- Configurable trade off: overhead vs failover responsiveness
  - δ_1 : Heartbeat requests generation timeout
  - δ_2 : Heartbeat reply timeout before the port is declared down
- Guaranteed max detection delay:
  - $\delta_1 + \delta_2$

Fast Reroute

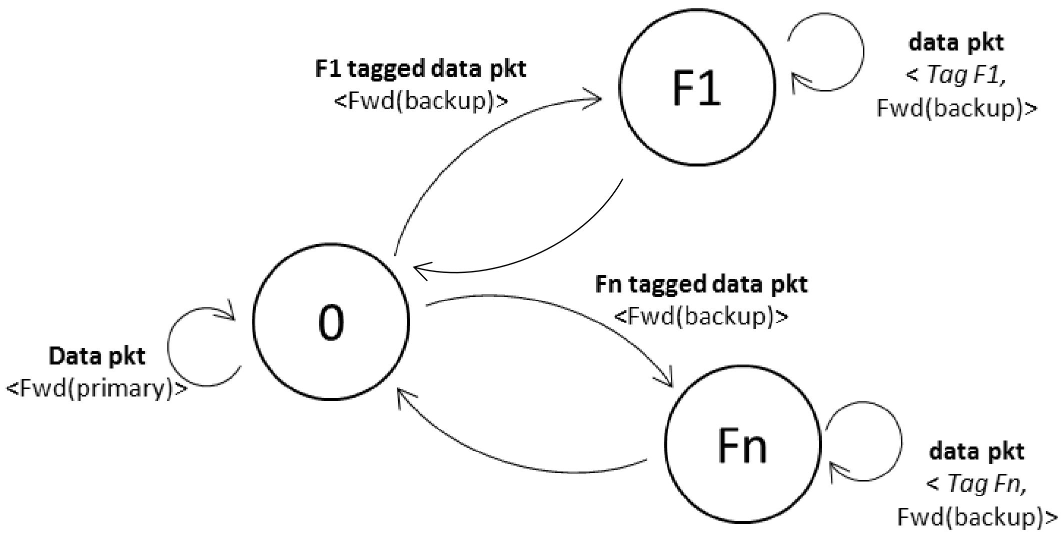


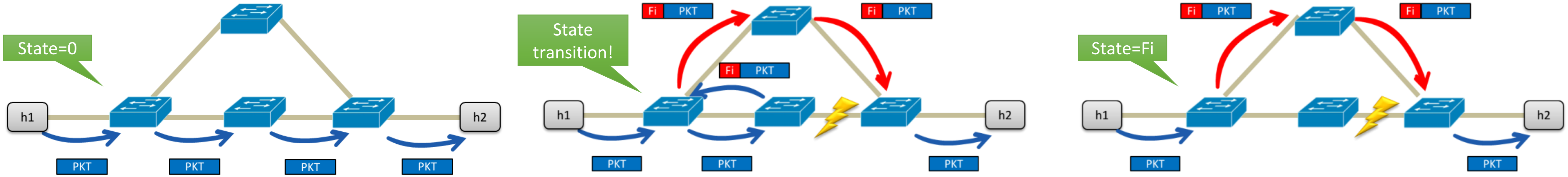
- MPLS label used to distinguish between different forwarding :
  - No tag → forward packet on the primary path
  - tag=Fi → forward packet on the detour for the i-th failure
- Zero losses after failure detection
- No controller intervention for all pre-planned failures
- What if no local alternative path is available?

Fast Reroute (2)

- Packets are tagged and bounced back up to a proper redirect point
- Tagged packets trigger a state transition:
  - updating the routing of the involved connections
- Still zero losses after failure detection!
- Tagged data packets as signalling
- No controller intervention!

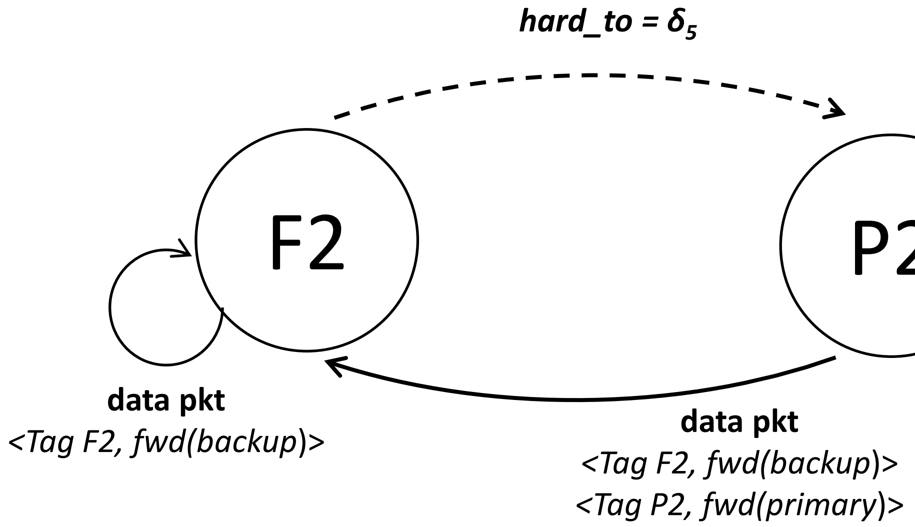
Lookup-scope = [ETH_SRC,ETH_DST]
Update-scope = [ETH_SRC,ETH_DST]

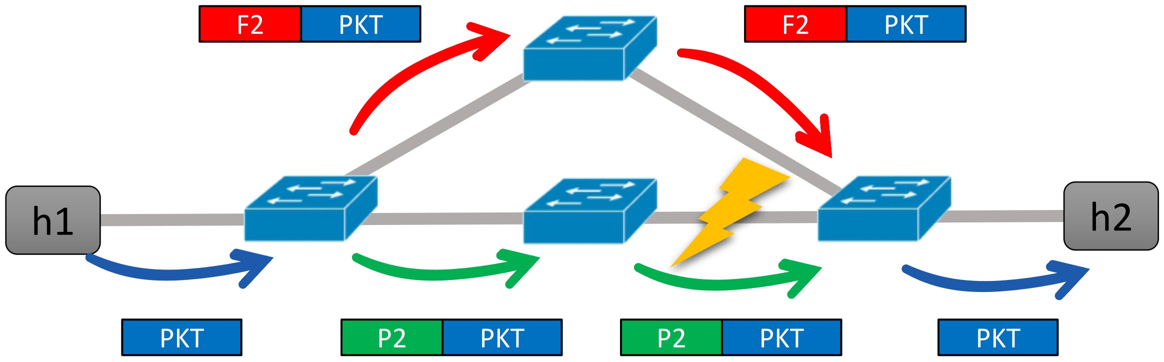




Path probing

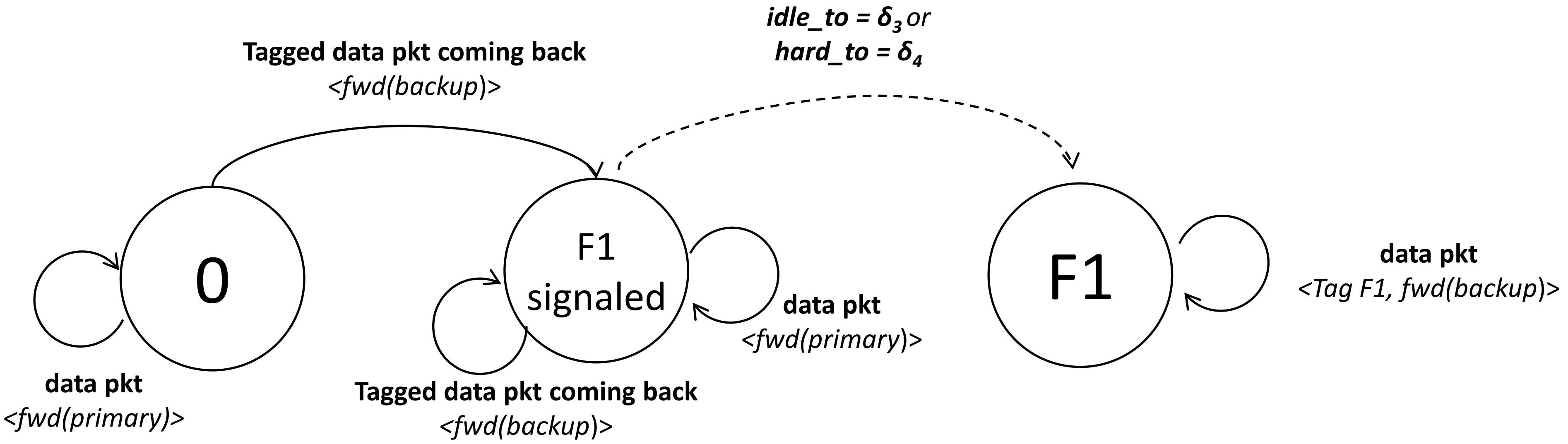
- How to restore the forwarding on the primary path?
- Programmable periodic probing for primary path availability



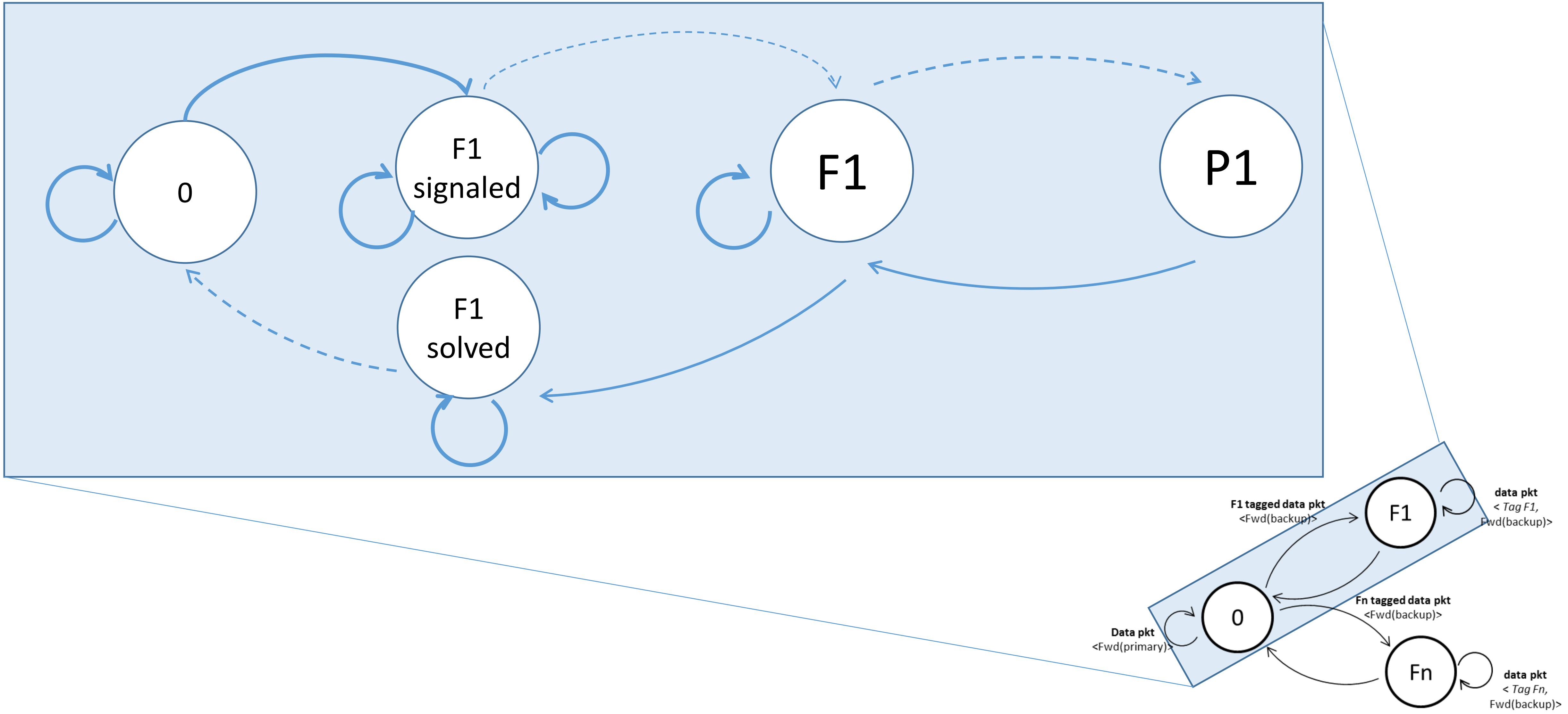


Flowlet-aware rerouting

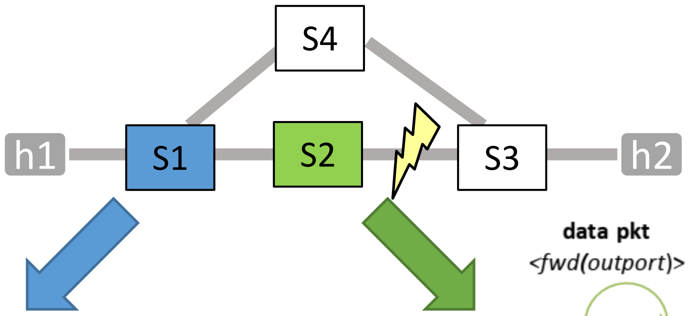
- Failover activation/deactivation can be post-poned
  - In order to minimize out-of-sequence, packets are kept on the primary path up to expiration of a burst of packets
  - Programmable idle timeout/hard timeout



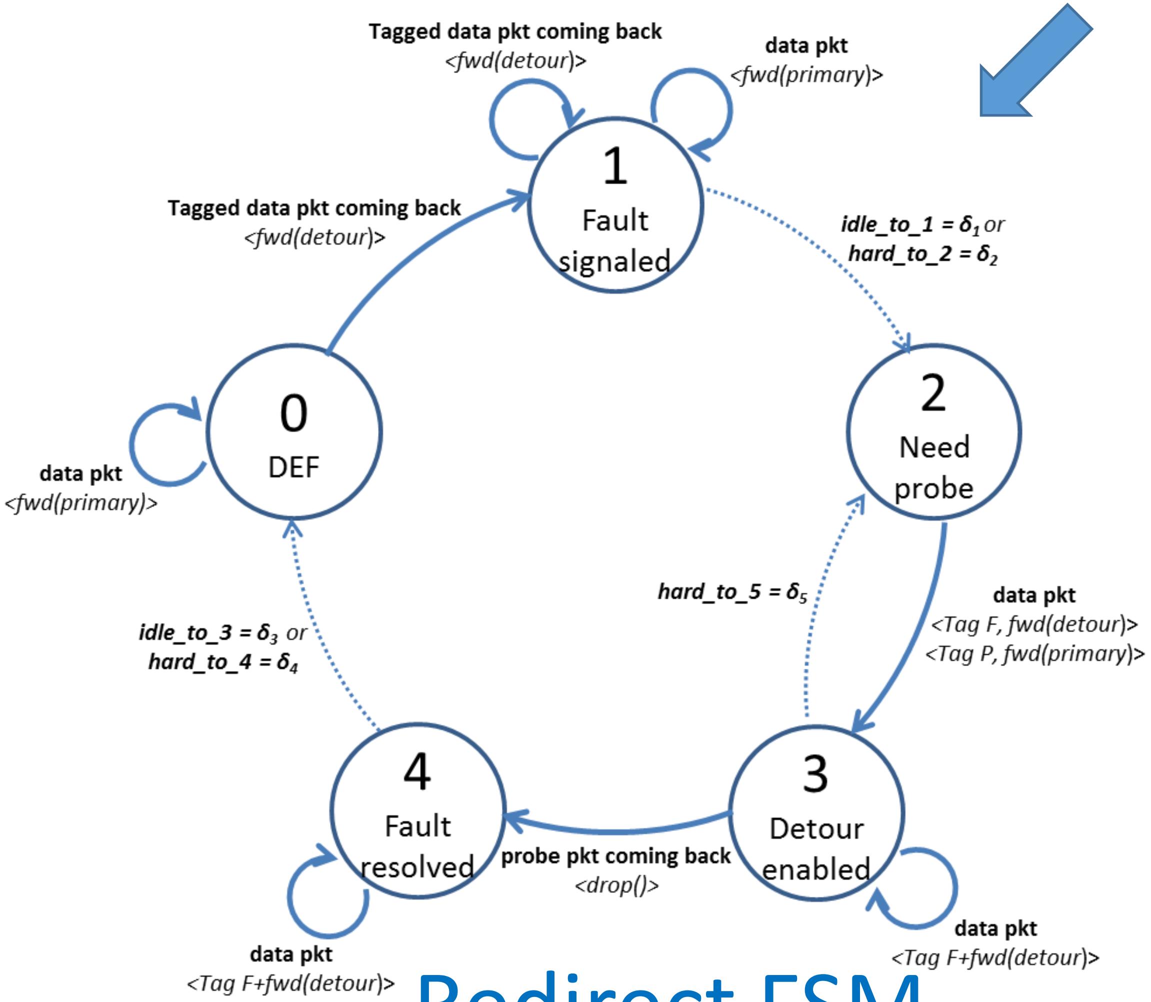
Putting all together: Fast reroute FSM



FSM: NO backup path is locally available

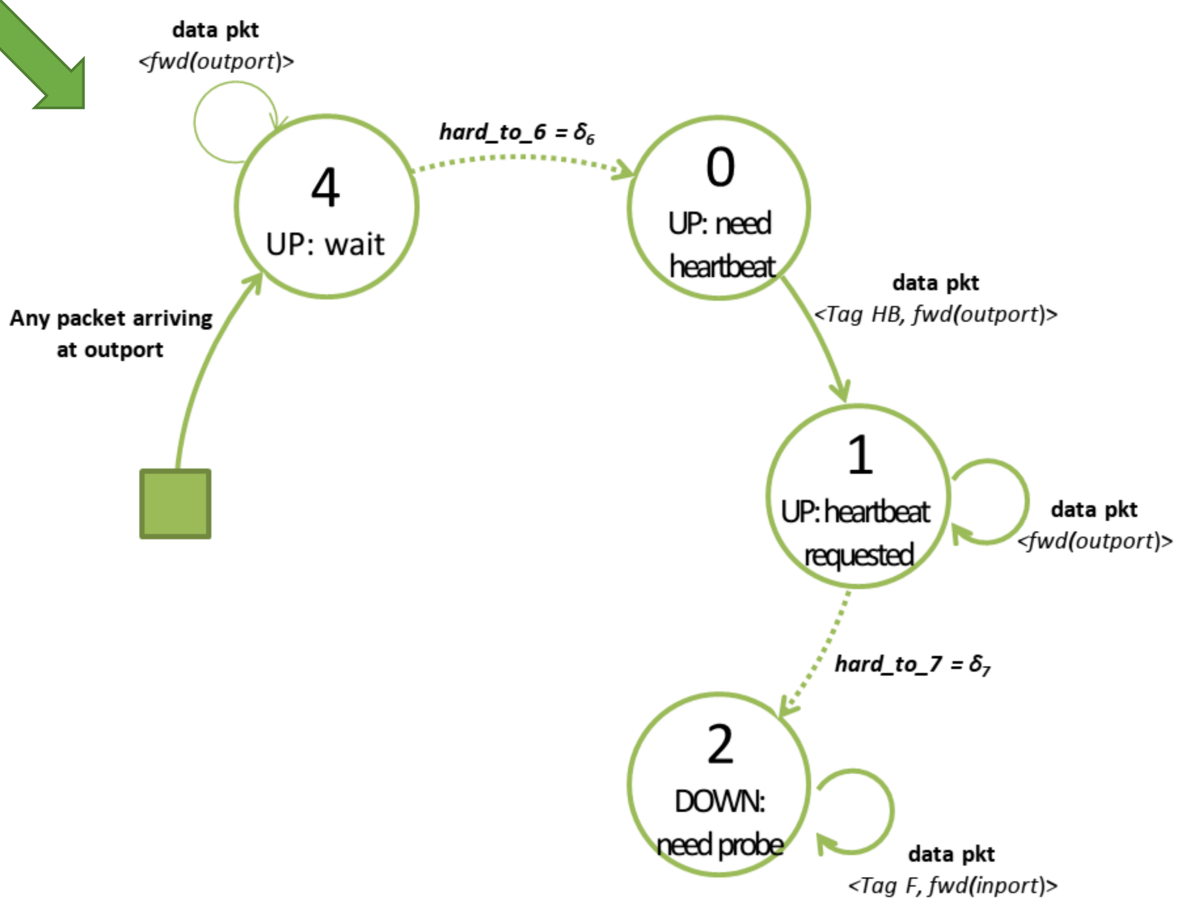


Lookup-scope = [ETH_SRC,ETH_DST]
Update-scope = [ETH_SRC,ETH_DST]



Redirect FSM

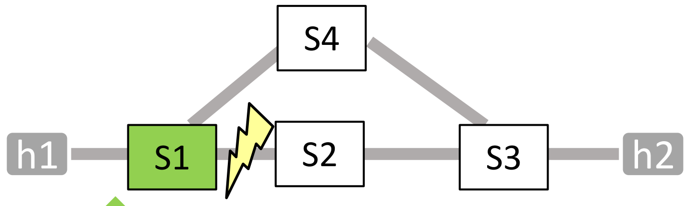
Lookup-scope = [METADATA]
Update-scope = [METADATA]

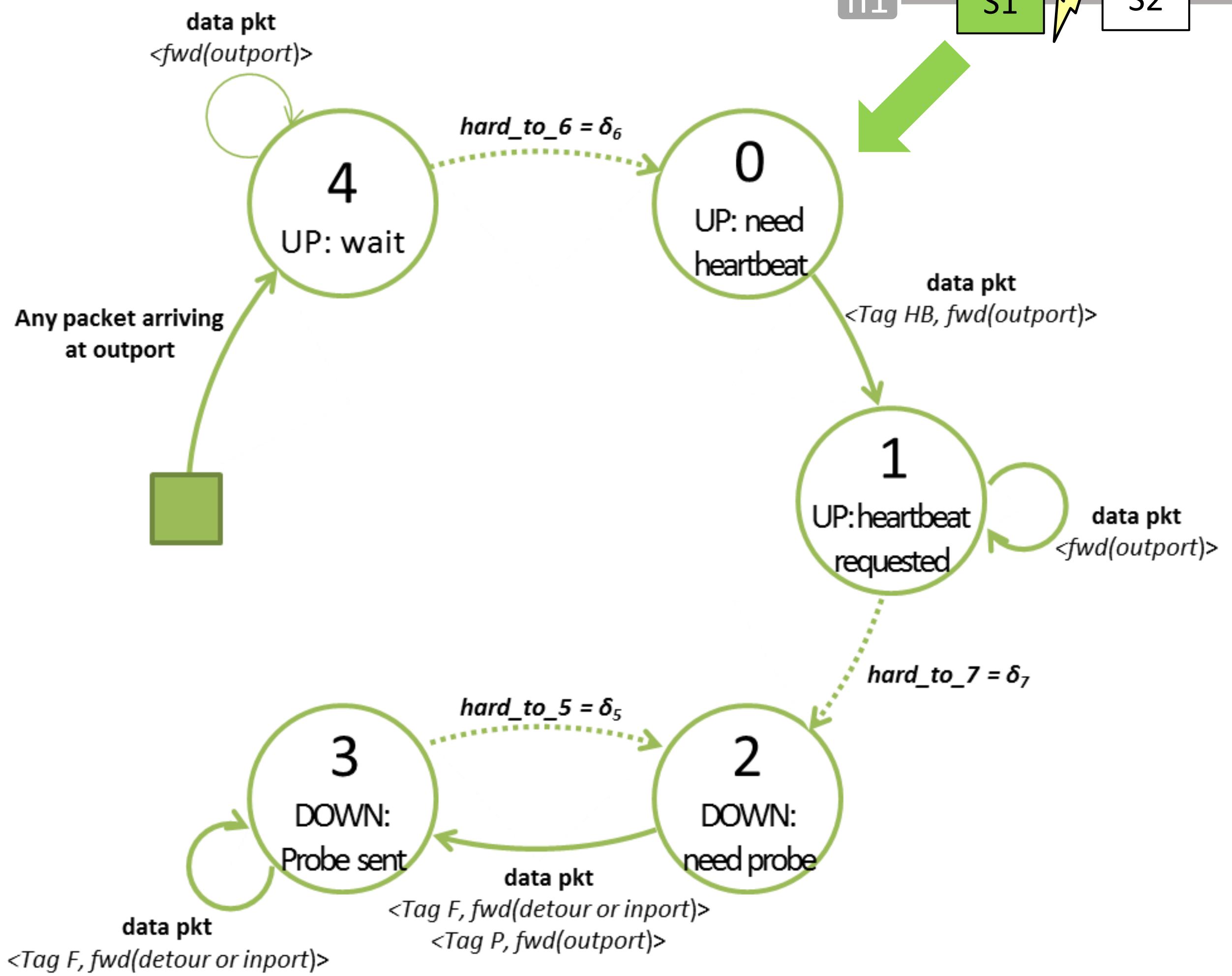


Failure Detect FSM

FSM (2): backup path is locally available

Lookup-scope = [METADATA]
Update-scope = [METADATA]





Redirect&Failure Detect FSM

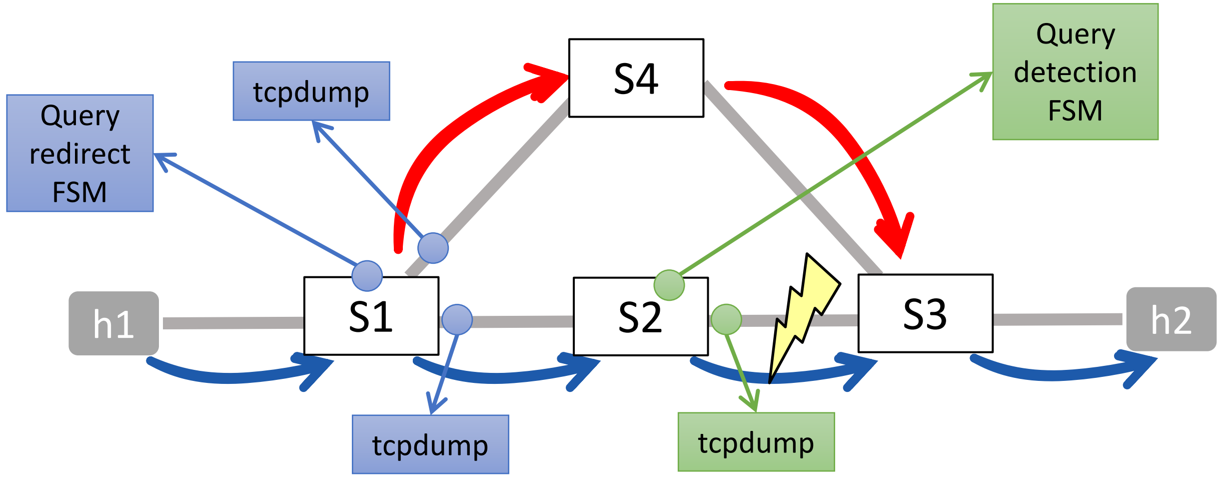
Software implementation

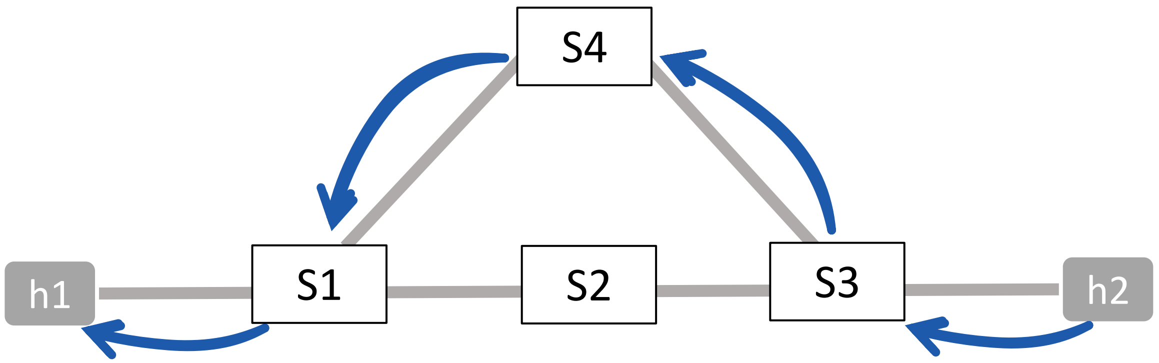
- OpenState
  - Ryu* controller
  - CPqD OpenFlow 1.3 softswitch*
  - https://github.com/OpenState-SDN/spider

- P4$_{14}$ based on openstate.p4 library
  - https://github.com/OpenState-SDN/openstate.p4

*modified with OpenState support http://openstate-sdn.org

Testbed

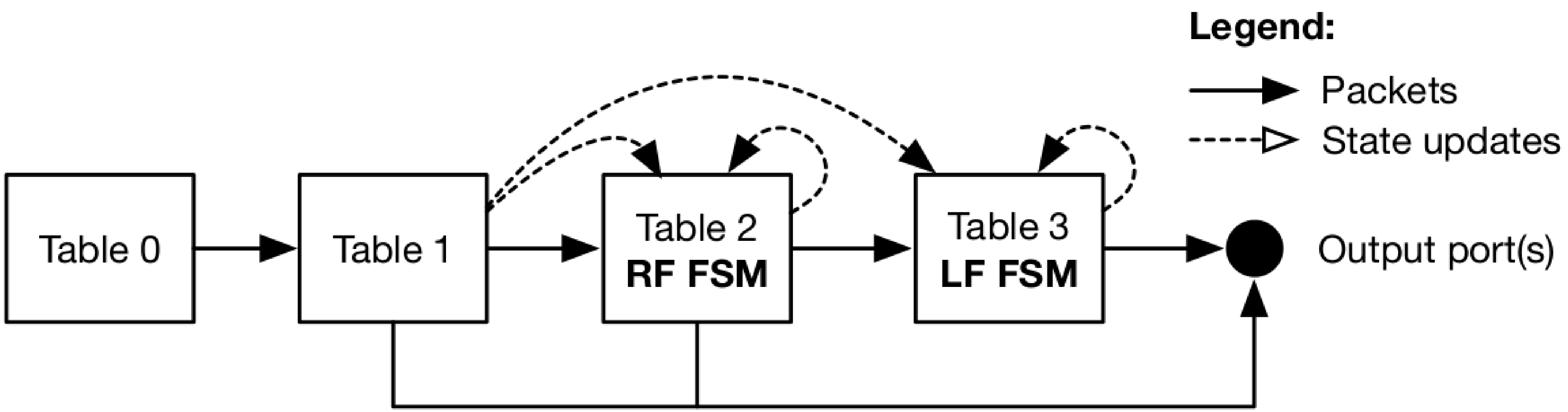




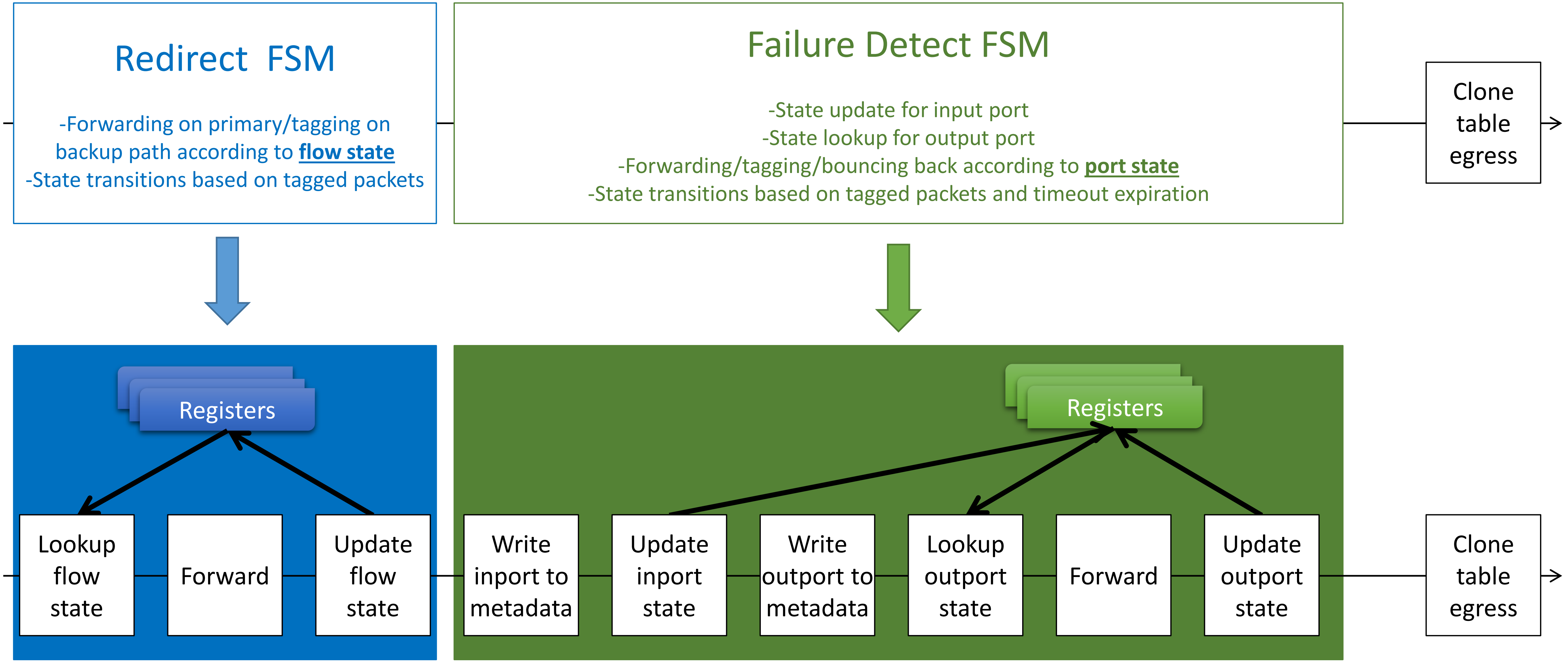
- Testing ping from h1 to h2
  - h1→h2 forwarded on path S1-S2-S3
  - h2→h1 forwarded on path S3-S4-S1
- Failure of link S2-S3
  - Backup path S1-S4-S3
  - Checking Heartbeat mechanism for failure detection
  - Checking Probe mechanism for primary path availability

VLAN tag	forwarding
16	Normal
17	Failure
20	HB_request
21	HB_reply
22	Probe

OpenState pipeline



P4 pipeline


Redirect FSM
-Forwarding on primary/tagging on backup path according to **flow state**
-State transitions based on tagged packets
Failure Detect FSM
-State update for input port
-State lookup for output port
-Forwarding/tagging/bouncing back according to **port state**
-State transitions based on tagged packets and timeout expiration
Clone table egress
Registers
Lookup flow state
Forward
Update flow state
Registers
Write inport to metadata
Update inport state
Write outport to metadata
Lookup outport state
Forward
Update outport state
Clone table egress

P4 pipeline (2)