Week14

For most controller applications, HyperFlow only requires minor modifications: they must dynamically tag events which affect their state. However, some of them must be further modified to ensure correct operation under temporal event reordering and transiently conflicting controller views, and guarantee scalability. Be- sides, we discuss how the controller applications must be modified to enable interconnection of independently- managed OpenFlow networks.

Event reordering: In HyperFlow, correct operation of control applications must not depend on temporal ordering of events except those targeting the same entity (e.g., the same switch or link), because different controllers perceive events in different orders. Besides, resilience to network partitioning requires control applications to tolerate out-of-order event delivery (even lagging several hours) without sacrificing correctness, because each partition is notified of the state of the other partitions upon reconnection.

Correctness: Transient inconsistencies among controllers may lead to conflicting decisions. To ensure correct operation in all cases, control applications must forward requests to the authoritative controller. The authoritative controller for a given flow is the one managing the flow’s source switch. Consider the switching/routing applications as an example: To ensure loop-free for- warding, flow paths must be set up by the controller managing the flow’s source switch. Other controllers must redirect the request to the authoritative controller

in case they receive a flow initiation event. As another example, consider a network with a policy which re- quires both the forward and reverse paths of all flows

to match. To guarantee this, the source controller must simultaneously set up both paths upon flow initiation. This modification ensures that the policy is always correctly enforced from the source controller’s perspective. Bounded number of possibly effective events: The number of events which possibly affect the state of a HyperFlow-compliant application must be bounded by O(h+l+s), where h is the number of hosts, l is the number of links, and s is the number of switches in the network. In other words, applications whose state may be affected by O(f(n)) events, where f(n) is any function of the number of flows in the network, incur a prohibitively large overhead and must be modified.