Flow-level State Transition as a New Switch Primitive for SDN

(HotSDN’14)

Masoud Moshref, Apoorv Bhargava, Adhip Gupta, Minlan Yu, Ramesh Govindan
Motivation

Current practice
- Proactive needs a priori knowledge
- Reactive has high delay

Opportunity: Local state is enough for many policies (stateful firewall, FTP monitoring, large source IP detection)

Key idea: State machine is a general but efficient abstraction to allow dynamic actions at switches
FAST (Flow-level State Transitions) Abstraction

- Controller proactively programs state transitions and actions at switches
- Switches run state machines and actions of a state

Examples:
- **Stateful firewall**: TCP state machine with actions that drop uninitiated flows
- **FTP Monitoring**: Track the states of control channel & allow data channel traffic
- **Large source IP detection**: Keep a counter per IP and compare it against a threshold
Controller translates state machines to switch API

FAST Control Plane

Switch agent

Switch agent

FAST compiler

Network controller

None

Init1

Init2

Est

Close 1

Close 2

SYN

SYNACK

ACK

FIN

FINACK
FAST data plane is implementable in hardware switch components.
Delay of going through all TCP states for FAST is small
1 packet, 1 flow: FAST: 28x faster (3ms)
> 64 concurrent flows: 6ms

FAST state lookup has small overhead:
Iperf throughput (Gbps): <5% overhead