: a DSL for Active Networking

Danny Yang

CS 6114 Final Project

Source: https://github.com/yangdanny97/p4-stacklang

Active Networking

- programmable packets: contrast with SDN and programmable switches
- packets include code that is executed on the network
- code can read information about packets/switches/links and modify how the network behaves

Implementation Overview

- Implemented as a switch in P4
  - Custom headers
  - Bytecode interpreter
- Mostly independent of hardware/target, aside from reading/writing metadata fields
- Minimal topology requirements (switches need to be able to forward packets to themselves)

Packet Format

metadata	program data	instructions x32	pre-allocated stack x32

- metadata - metadata copied into this header when packet first arrives at a switch
- program data - PC, SP, done/error flags, etc
- instructions - representation has single int32 operand per instruction (set to 0 for instructions that don't have one)
- stack values - values are int32

Packet Size Overhead: 38 bytes + 4 bytes per stack size + 5 bytes per instruction

(326 bytes in current implementation)

Instruction Set

- Inspired by bytecodes and stack based languages
- Around 3 dozen instructions
- Programs can access 2 types of memory:
  - stack (carried on the packet)
  - registers (persistent state stored on each switch, not shared)
- Special instructions for reading metadata and setting egress port

Execution Model

- Program is executed once on each hop between source and destination
- Ingress executes a single instruction; packet forwarded back to self for next instruction
- When program is done, the packet is forwarded to next hop

Example Programs

- General computation: Fibonacci/Factorial
- Source Routing
- Packet Drop Detector
- Simulated Match Action Table

Tradeoff of program complexity vs packet size overhead

Room for Improvement

- Contention between multiple programs being executed by the same switch concurrently
    - only applies to programs that access registers (any number of stack-only programs can be executed concurrently without issues)
- Not entirely hardware independent
- Current implementation is not modular