

Millions of Little Minions: Using Packets for Low Latency Network Programming and Visibility

David Eschbach

March 02, 2018

Introduction

Why minions

- Minions: Tiny Packet Programs (TPP)
- Previous work: *Tiny Packet Programs for low-latency network control and monitoring* (HotNets'13)



Introduction

Addressed Problems:

- Poor visibility of what really happens inside the network
- Introducing new dataplane functionality is expensive

Proposed solution:

- Use TPPs to give end-hosts access to network state through the dataplane

Goal:

- Show that TPPs are useful and feasible

Tiny Packet Programs

Definition and Properties

Definition: A TPP is any Ethernet packet with a uniquely identifiable header that contains instructions, some additional space and an optional encapsulated Ethernet payload.

- Programmable interface giving end-hosts access to switch memory
- Small programs located in packet headers
- General Approach: Refactor network tasks into:
 1. TPPs that are executed inside switches
 2. Programs at end-hosts analyzing the results of the TPPs

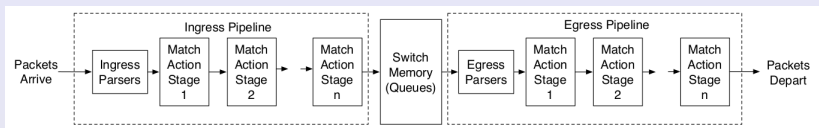
Running Example: Micro-burst Detection

- Problems today:
 - Observation mechanisms work at (too) slow rates
 - Which queues would you monitor?
- Using TPPs we can:
 1. query for queue sizes of links
 2. analyze results in end-host

```
PUSH [Queue: QueueOccupancy]  
PUSH [Switch: ID]
```

Tiny Packet Programs

Implementation in Switches

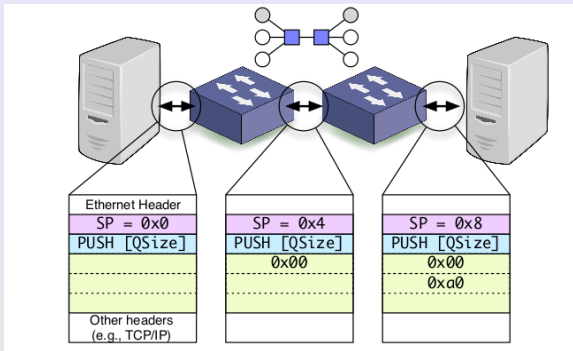


- TCPU: One execution unit per stage
- TPP compiler at end-host decides executing state for each instruction

Running Example: TPP Compiler knows at which stage queue size information is available

Tiny Packet Programs

Running Example: Execution in Dataplane



- Note: Maximum number of hops known in advance.

Tiny Packet Programs

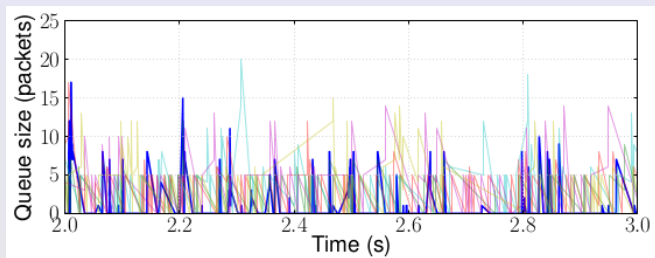
TPP control plane

- central entity in network
- provides interface to insert TPPs
- responsible for access control policies

Tiny Packet Programs

Implementation and execution in End-hosts

- control plane agent and dataplane shim
- Running example:



More Usages of TPPs

Rate-based Congestion Control

- RCP: Rate Control Protocol
- Congestion control algorithm that informs end-hosts about minimal link capacity
- RCP* Phases:
 1. Collection of link rates using TPPs
 2. Computation of new rates for each link at end-host
 3. update of link rates using TPPs

More Usages of TPPs

Distributed Load balancing

- CONGA: scheme for traffic load balancing that uses information about congestion stored in switches
- Multipath routes must be available
- CONGA*:
 1. Use TPPs to get link utilization across different routes.
 2. End-hosts select the path for each flowlet.

More Usages of TPPs

- Low overhead measurement
- Network Troubleshooting Frameworks
- Network verification
- Fast network updates

Comparison with Related Work

- Realization of Active networks
- Reduced instruction set
- Sprocket
- Generic approach of accessing switch state

Results & Limitations

Results:

- Usefulness of TPPs shown by presenting examples
- Feasibility of TPP support shown by implementation in software and tests on real hardware

Restrictions for solving tasks with TPPs:

- small instruction set
- only end-hosts can trigger tasks

Discussion & Questions

- Overhead considerations
- Writes to network state
- A TPP needs to fit into MTU

Questions?