Implementing a Monitoring Device using a P4 Programmable Switch

Samuel Dao, Mark-Anthony Box
Advisors: Ali Mazloum, Ali AlSabeh

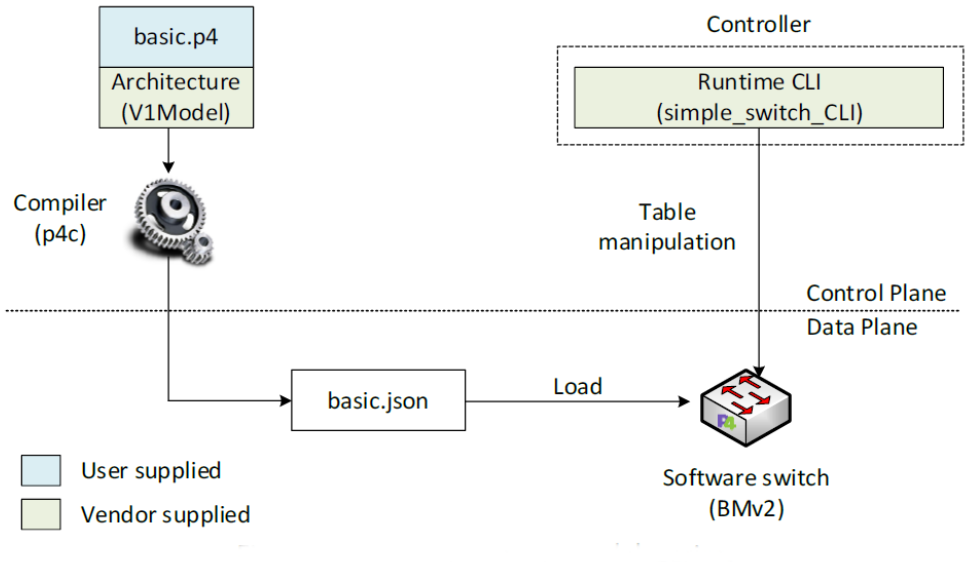
Department of Integrated Information Technology
University of South Carolina

May 2022

Agenda

- Introduction
- Background Information
- Research Challenge
- Solution
- Conclusion

Introduction



Network devices allow hardware on a network to communicate. They **process** packets and route them to the proper destination.

Typically, only vendors have had **control** over network design and functionality in devices.

P4 is a new programming language that specifies how data plane devices, such as programmable switches and routers, **process** packets.

Background Information

Current monitoring devices do not provide enough **granularity** to fix network issues.

P4 gives **total control** to developers. Implementing changes now takes **minutes** instead of years.

P4 utilizes programmable **ASIC** chips that control the functionality of the switch.

Research Challenge

Goal: Use the capability of P4 devices to monitor and track flows and collect corresponding statistics.

- Source and destination IP addresses
- Source and destination transport layer ports
- Amount of the unidirectional traffic (bytes)
- Bit rate per second
- Flow start and end time

Solution

Using P4, we can **record** statistics of packets at line-rate without adding processing overhead.

Packets will be uniquely identified by **flows**. Flows are packets that contain the same information and belong to the same **stream** of data.

In this project, **flows** are defined as packets that contain the same: Source and Destination IP Addresses, same Source and Destination Ports, and the same IP protocol.

Solution cont.

Source IP address, destination IP address, source port, and destination port are **extracted** from the packet's headers.

Unidirectional traffic is measured at the data plane level through **summing** incoming packets length

Bit rate is calculated by dividing the flow's size by the duration of the flow

Flow start and end times are extracted from the P4 switch's standard **meta-data**

Unique **Flow IDs** will be generated using a hashing algorithm.

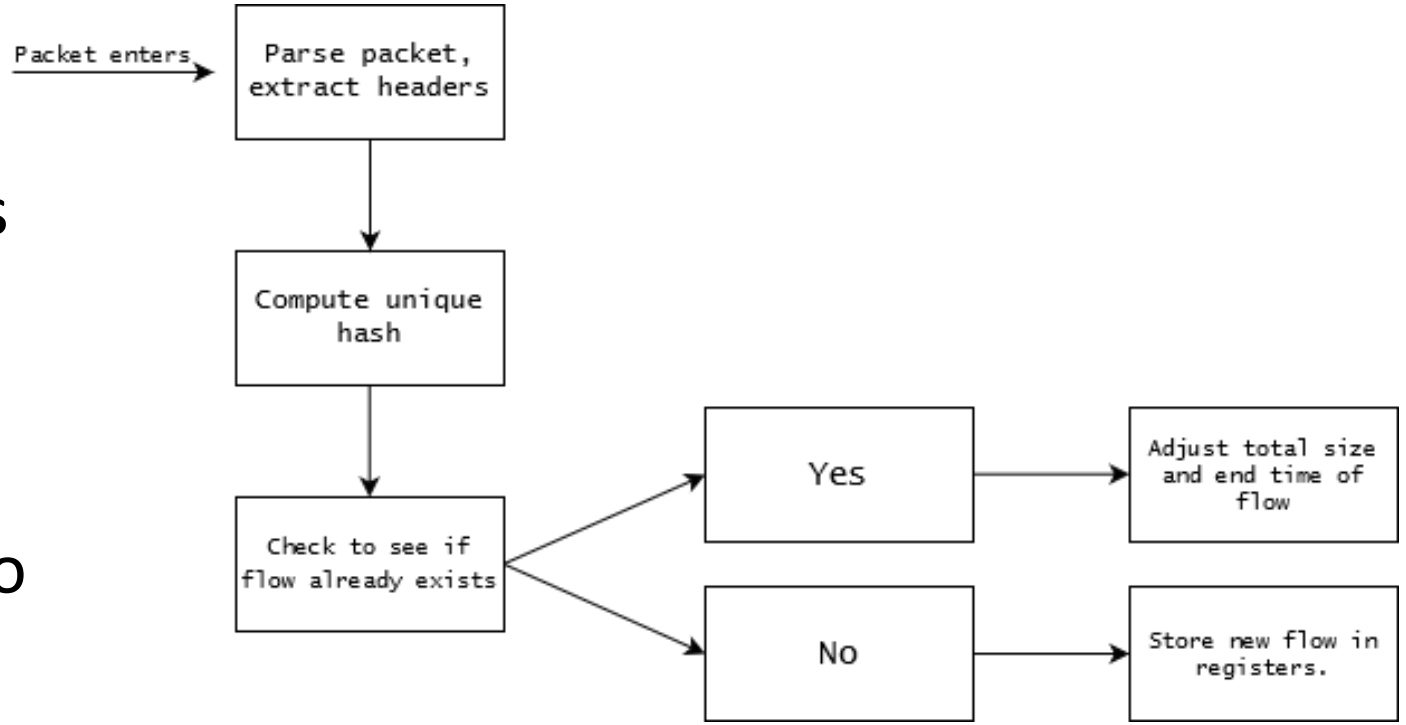
Solution cont.

Statistics of each **flow** are stored in a dedicated registers

We designed our own python script that can **extract** the values of the registers from the data plane

The python script is connected to the data plane through **APIs** provided by the vendors

High-level Overview of Workflow



Conclusion

Using P4, we were able to successfully pull the targeted information, from the **headers** of packets and metadata of the switch and store them in registers.

We can uniquely classify flows by calling a **hashing algorithm** against our targeted statistics.

We are then able to pull these values from the registers during **runtime,** and format them using a python script.