

CWM Programmable Networks

Exercise 4: In-network Calculator

Introduction

This exercise requires implementing a basic calculator using a custom protocol header written in P4. The header will contain an operation to perform (e.g., addition, subtraction) and two operands. When a switch receives a calculator packet header, it will execute the operation on the operands, and return the result to the sender.

A skeleton P4 program named `calc.p4` is located under CWM-ProgNets/assignment4, and initially drops all packets. Your task is to extend it to properly implement the calculator logic.

You need to submit the completed exercise via Canvas.

Setup

In this exercise you will use two machines: your lab machine, and a Raspberry Pi, connected directly using a single Ethernet cable. The cable should be connected from the lab machine's USB adapter to the Raspberry Pi's Ethernet port (no adapter).

The test sequence is the same through all parts of this exercise:

- Login to the raspberry Pi.

```
ssh pi@192.168.10.2
```

- Compile the P4 program `calc.p4` under `CWM-ProgNets/assignment4`

```
p4c --target bmv2 --arch v1model --std p4-16 calc.p4
```

- Run the compiled program.

```
sudo simple_switch -i 0@eth0 calc.json
```

- On two separate terminal windows, run the python scripts provided under `CWM-ProgNets/assignment4` (each script on a different window):

```
sudo python calc_sender.py
```

```
sudo python calc_receiver.py
```

- Enter an equation and check if you get the correct results.

Stage 1

The P4 skeleton `calc.p4` is located under `CWM-ProgNets/assignment4`.

It initially drops all packets.

Two Python-based driver programs, `calc_sender.py` and `calc_receiver.py` allow you to test your calculator. The programs provide a new prompt, at which you can type basic expressions.

The test harness will parse your expression and prepare a packet with the corresponding operator and operands. It will then send a packet to the switch for evaluation. When the switch returns with the result of the computation, the test program will print the result.

Because the original calculator program is not implemented, you should see an error message testing it.

At this stage, you should run the test sequence noted above and check that you receive the expected error.

Stage 2

In this stage, you need to implement the calculator:

1. Define a custom calculator header.
2. Implement the switch logic to:
 - a. Parse the header
 - b. Perform the requested operation
 - c. Write the result in the header
 - d. Return the packet to the sender

Use the following header format (numbers indicate bytes, e.g., operand A will be in bytes 4-7).

0	1	2	3
P	4	Version	Op
Operand A			
Operand B			
Result			

Use the following conventions:

- P is an ASCII Letter 'P' (0x50)
- 4 is an ASCII Letter '4' (0x34)

- Version is 0.1 (0x01)
- Op is an operation to Perform:
 - '+' (0x2b):
Result = OperandA + OperandB
 - '-' (0x2d):
Result = OperandA - OperandB
 - '&' (0x26):
Result = OperandA & OperandB
 - '|' (0x7c):
Result = OperandA | OperandB
 - '^' (0x5e):
Result = OperandA ^ OperandB

Assume that the calculator header is carried over Ethernet, and use Ethernet type 0x1234 to indicate the presence of the header.

Your task is to implement only the P4 calculator program. There is no control plane logic, so only data plane implementation is needed.

Submit on Canvas your p4 file (calc.p4).