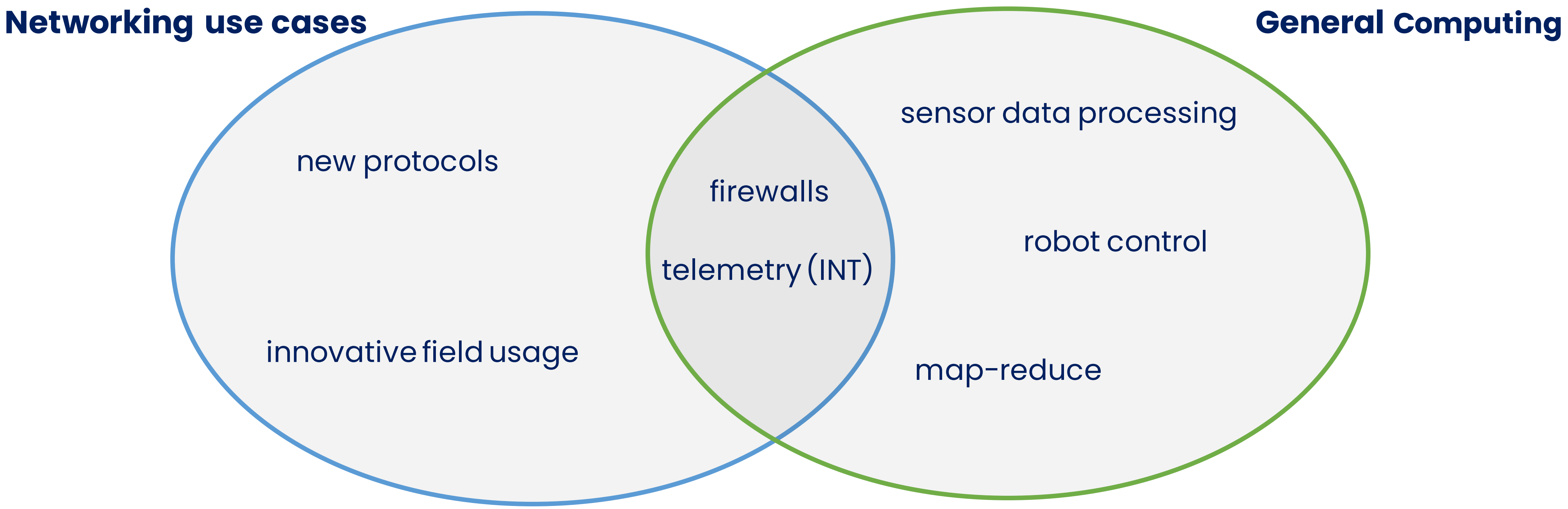
P4 2022 Workshop
May 24-26th

P4RROT: Generating P4 code for the Application Layer

Csaba Györgyi, Sándor Laki, Stefan Schmid

Where to use P4?



Disclaimer: P4 is great, and polarbears are cute. We all love P4. The challenges we are about to show are because of the slightly *out of scope usage* of the language.

Challenges

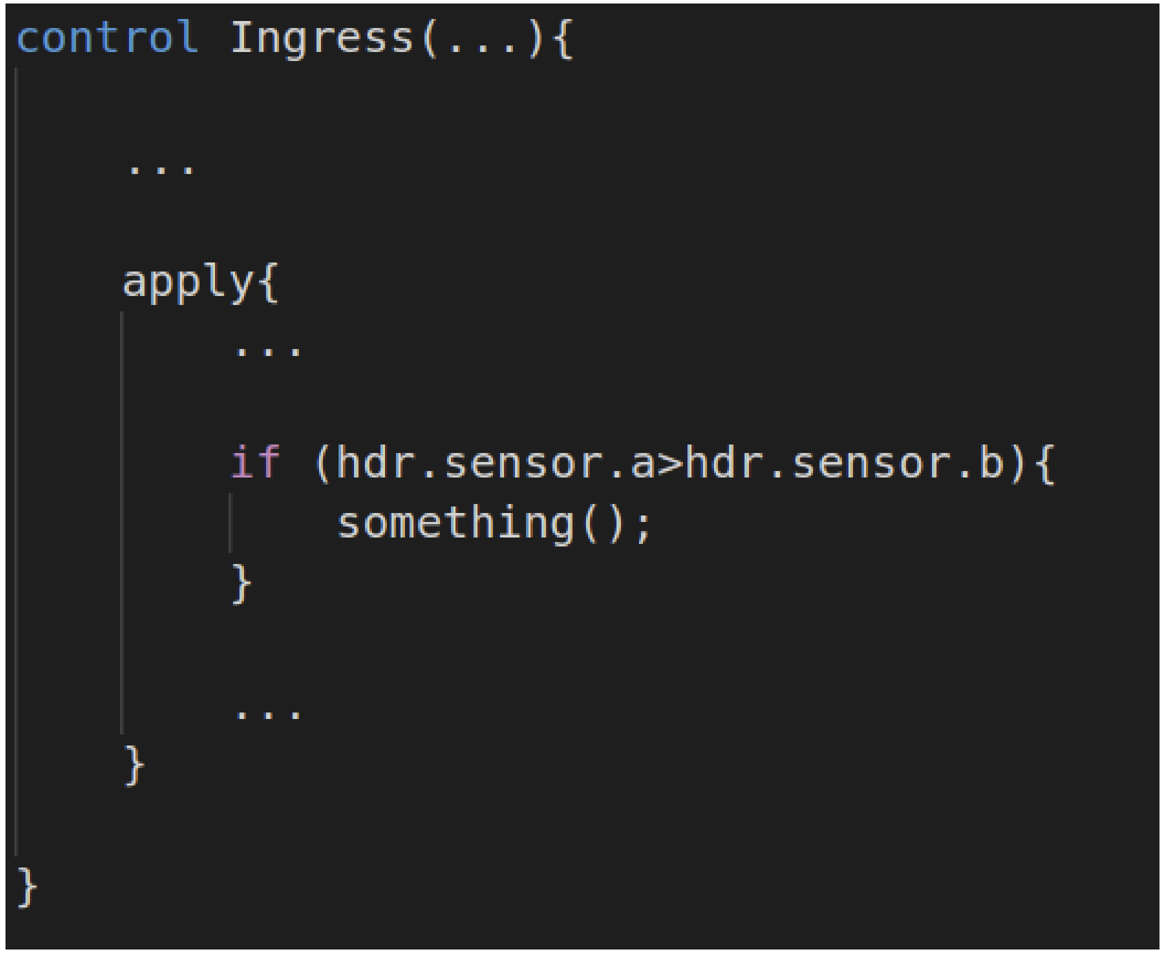
- Boilerplate code
- Cross-layer dependencies
- Lack of high-level encapsulation
- Tricks and workarounds
- Implementation details become design decisions
- Fragmented code
- Hard to test

A simple example (specification)

- If a is greater than b, then do something. Variable a and b are 64 bit long unsigned integers between 0 and 2^{50}.

```
…
if a>b:
    something()
…
```

A simple example (trivial way)

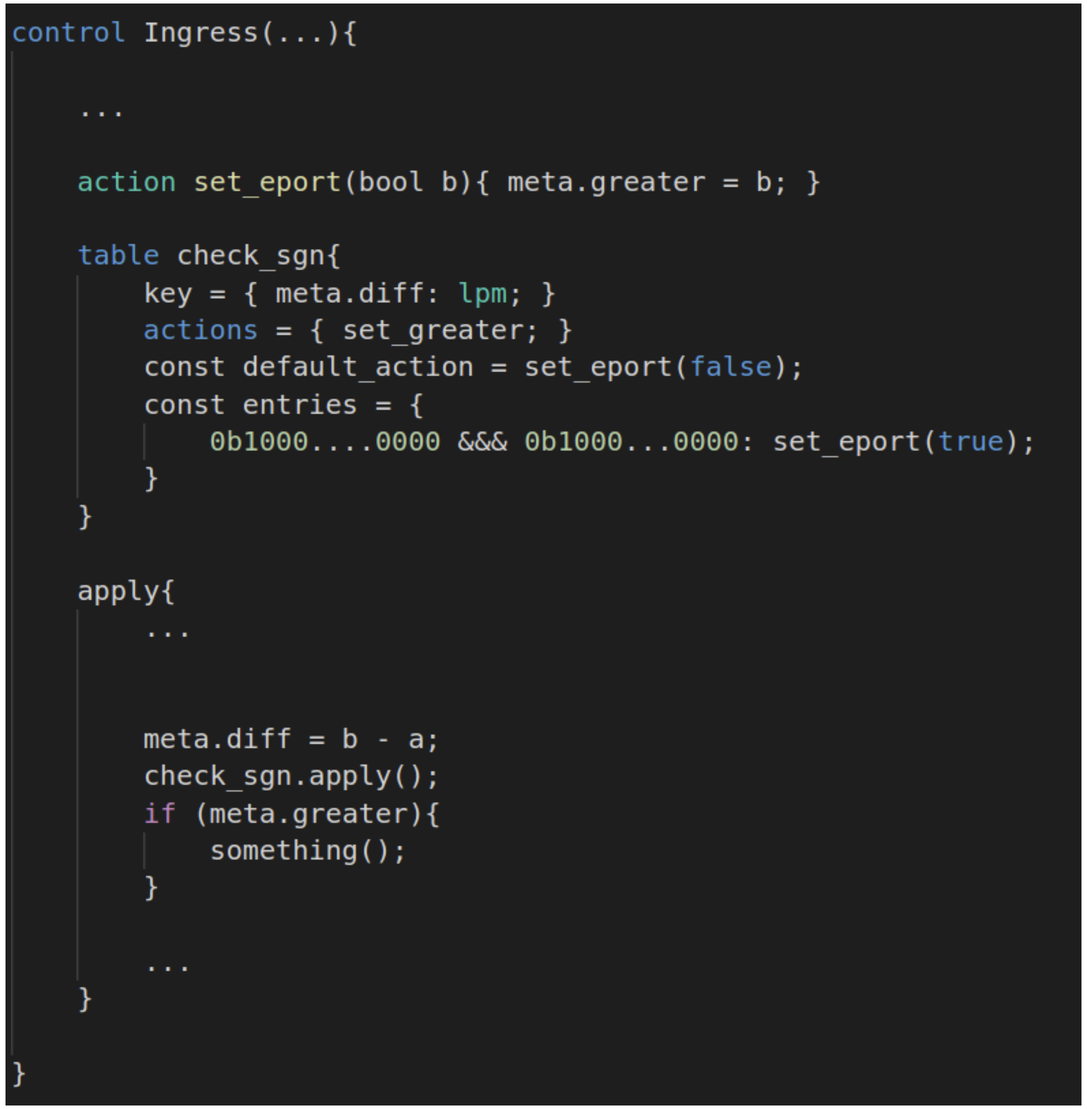


```
control Ingress(...){

    ...

    apply{
        ...

        if (hdr.sensor.a>hdr.sensor.b){
            something();
        }

        ...
    }

}
```

Compiler Error:

Sorry, my friend, you can not compare such large values…

A simple example (using LPM)



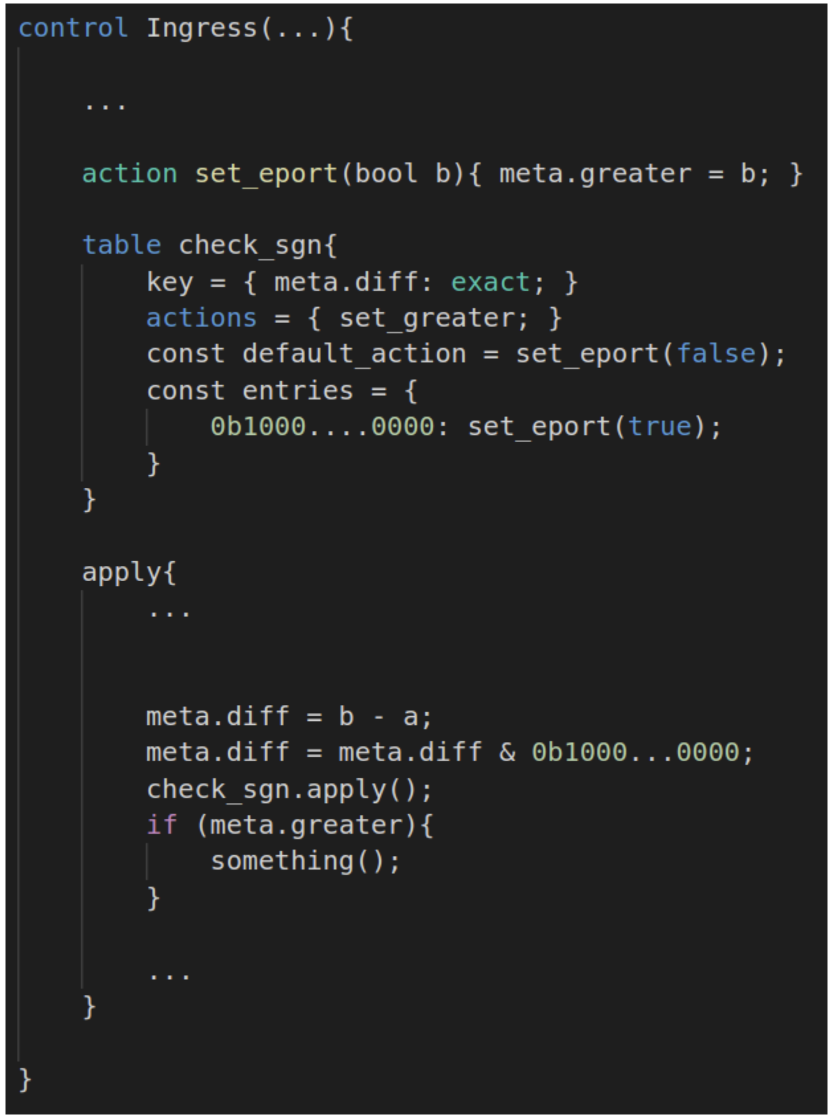
```
control Ingress(...){

    ...

    action set_eport(bool b){ meta.greater = b; }

    table check_sgn{
        key = { meta.diff: lpm; }
        actions = { set_greater; }
        const default_action = set_eport(false);
        const entries = {
            0b1000....0000 &&& 0b1000...0000: set_eport(true);
        }
    }

    apply{
        ...


        meta.diff = b - a;
        check_sgn.apply();
        if (meta.greater){
            something();
        }

        ...
    }

}
```

- Introducing an extra variable for the difference
- using longest prefix match to detect underflow

Compiler Error:

Sorry, boss, we are out of TCAM…

A simple example (using SRAM)



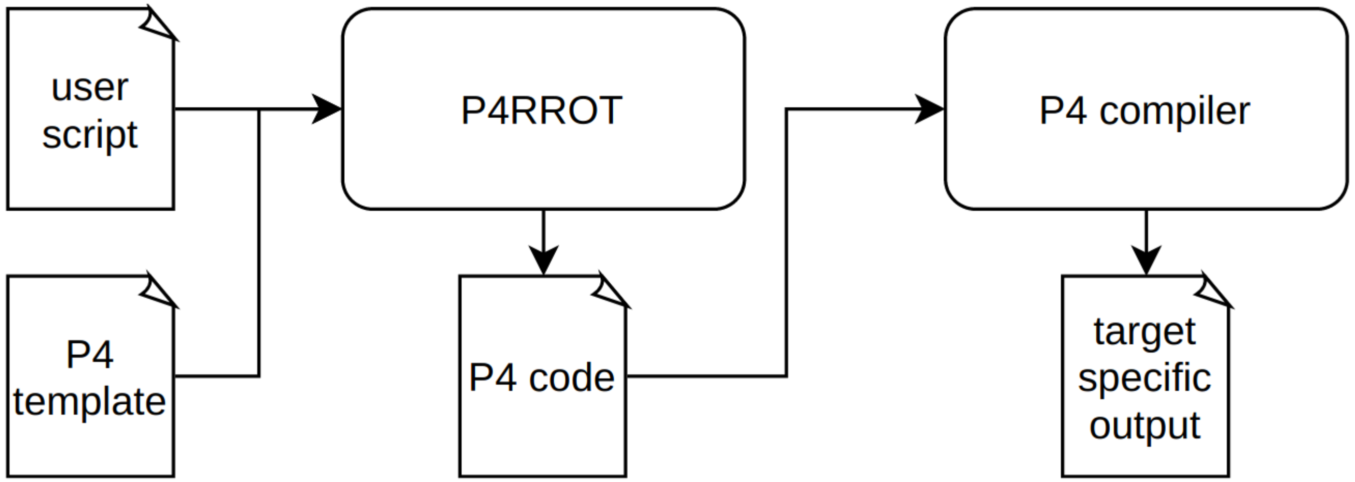
```
control Ingress(...){

    ...

    action set_eport(bool b){ meta.greater = b; }

    table check_sgn{
        key = { meta.diff: exact; }
        actions = { set_greater; }
        const default_action = set_eport(false);
        const entries = {
            0b1000....0000: set_eport(true);
        }
    }

    apply{
        ...


        meta.diff = b - a;
        meta.diff = meta.diff & 0b1000...0000;
        check_sgn.apply();
        if (meta.greater){
            something();
        }

        ...
    }

}
```

- mask the difference
- use an exact match

Compilation successful. :)

An open-source code generator

- Narrowing down the target use cases: application layer logic
  - Simplifications and assumptions. => Overcoming hindering factors
- Simple and familiar interface implemented in a high-level and well-known language (Python 3)
  - Easy to adopt
- Modularity, easy to extend
  - Flexibility
  - (Possibly provided as a service)



Example:
A number guessing game

- Input-Output like "declaration".
- Usage is similar to Keras, TensorFlow, LINQ, ….
- The generated code is easy to read and modify.

```
fp = FlowProcessor(
        istruct = [('guess',uint8_t)],
        locals  = [('l',bool_t),('good',bool_t),('solution',uint8_t)],
        ostruct = [('r1',uint8_t),('r2',uint8_t)],
        state   = [ SharedVariable('shared_solution',uint8_t) ]
    )

fp\
.add(Comment('init variables'))\
.add(ReadFromShared('solution','shared_solution'))\
.add(AssignConst('good',True))\
.add(AssignConst('r1',ord(':')))\
.add(AssignConst('r2',ord(')')))\
.add(Comment('check whether solution<guess'))\
.add(LessThan('l','solution','guess'))\
.add(If('l'))\
        .add(AssignConst('r1',ord('x')))\
        .add(AssignConst('r2',ord('<')))\
        .add(AssignConst('good',False))\
    .EndIf()\
.add(Comment('check whether solution>guess'))\
.add(GreaterThan('l','solution','guess'))\
.add(If('l'))\
        .add(AssignConst('r1',ord('x')))\
        .add(AssignConst('r2',ord('>')))\
        .add(AssignConst('good',False))\
    .EndIf()\
.add(Comment('generate a new number if required'))\
.add(If('good') )\
        .add(AssignRandomValue('solution',0,255))\
        .add(WriteToShared('shared_solution','solution'))\
    .EndIf()\
.add(Comment('send back the result'))\
.add(SendBack())


fs = FlowSelector(
        'IPV4_UDP',
        [(UdpDstPort,5555)],
        fp
    )

solution = Solution()
solution.add_flow_processor(fp)
solution.add_flow_selector(fs)
solution.get_generated_code().dump('test.p4app')
```

Defining inputs and outputs and other variables for the FlowProcessor

Populating processing steps

Channeling the proper packets to the FlowProcessor with the FlowSelector

Composing the parts of the solution.

Summary

- P4RROT is an open-source code generator speeding up offloading server functionalities by generating P4 code using a high-level API.
    - Fast prototyping, meant for the application layer
    - Reusing solutions
    - (Helps getting to know targats)

- https://github.com/Team-P4RROT/P4RROT
- https://arxiv.org/pdf/2204.02739.pdf

2022 Workshop
May 24-26th

Thank You

https://github.com/Team-P4RROT/P4RROT