TeraFlow SDN by ETSI

Interactive Session 2

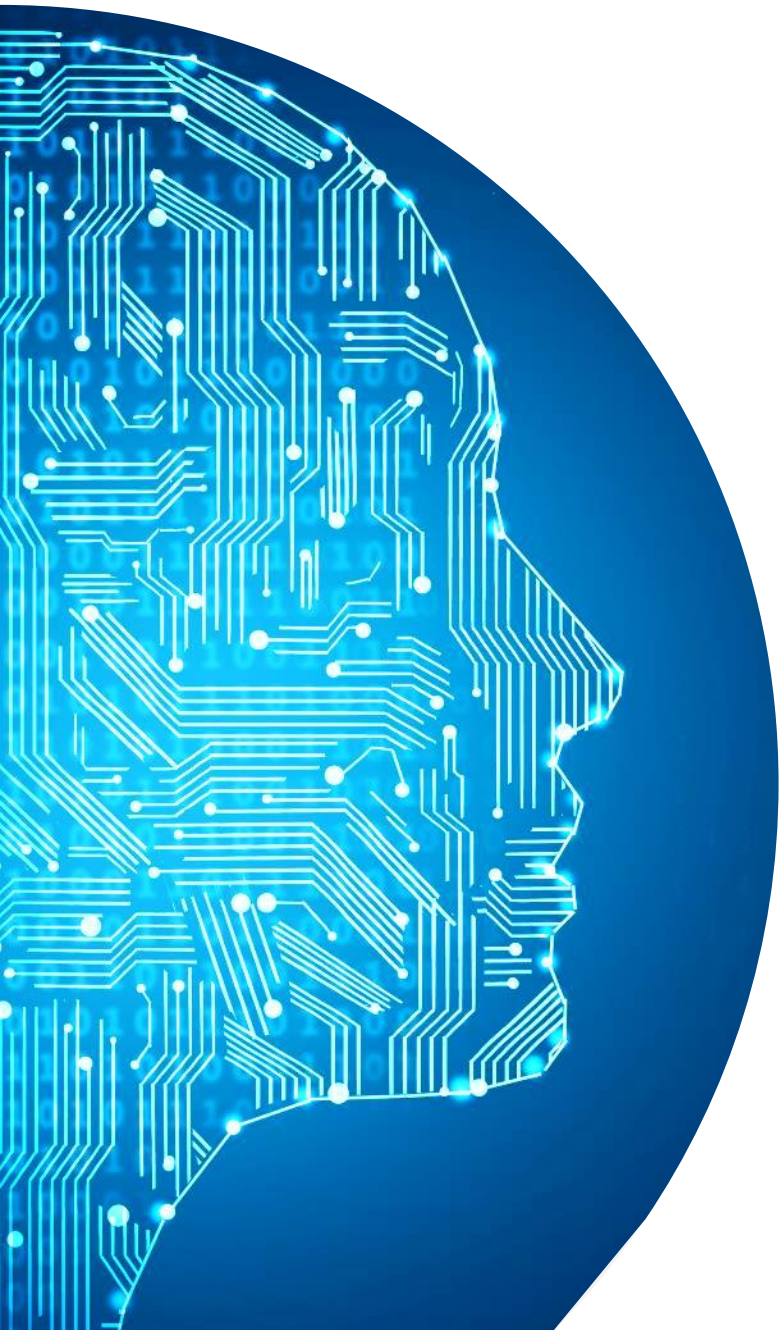
Panagiotis Famelis, Georgios P. Katsikas

ETSI TFS – Hackfest #3, October 17, 2023

TeraFlow SDN

Quick P4 tutorial

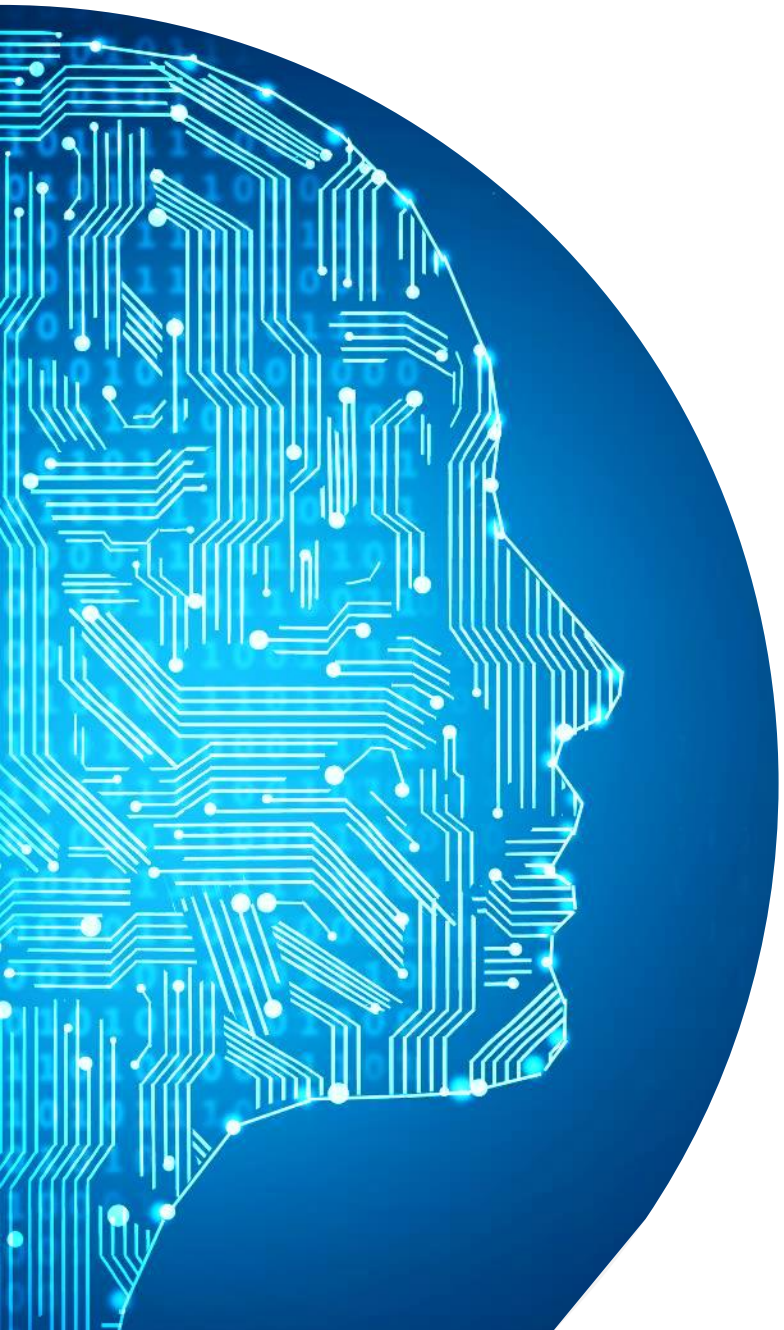
```
cd ~/controller/src/tests/hackfest3/p4
```



P4 program structure

A P4 program comprises of the following blocks:

- Header Definitions
  - Defining the protocol headers
- Parser
  - A state machine that dictates how to parse a packet
- Control Blocks
  - Comprising of Match-Action Tables
- Deparser
  - Defining how the packet will be encoded in the wire



Header definition and Parser

```
header Ethernet_h {
    bit<48> dstAddr;
    bit<48> srcAddr;
    bit<16> etherType;
}
```

```
parser MyParser (packet_in pkt, out accepted_packet hdr) {
    state start {
        pkt.extract(hdr.ethernet);
        transition select (pkt.ethernet.etherType) {
            0x800: parse_ipv4;
        }
    }
    state parse_ipv4 {
        pkt.extract(hdr.ip);
        transition accept;
    }
}
```

Control Block

- The main part of the program are the user-defined tables, comprising of:
  - A set of keys
  - A set of actions

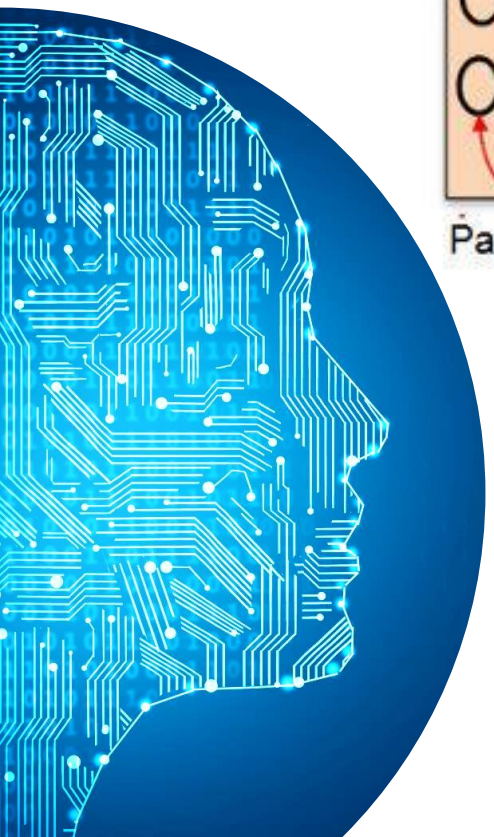
- The controller instructions are basically table entries consisting of:
  - A key
  - A corresponding action
  - Parameters for the action

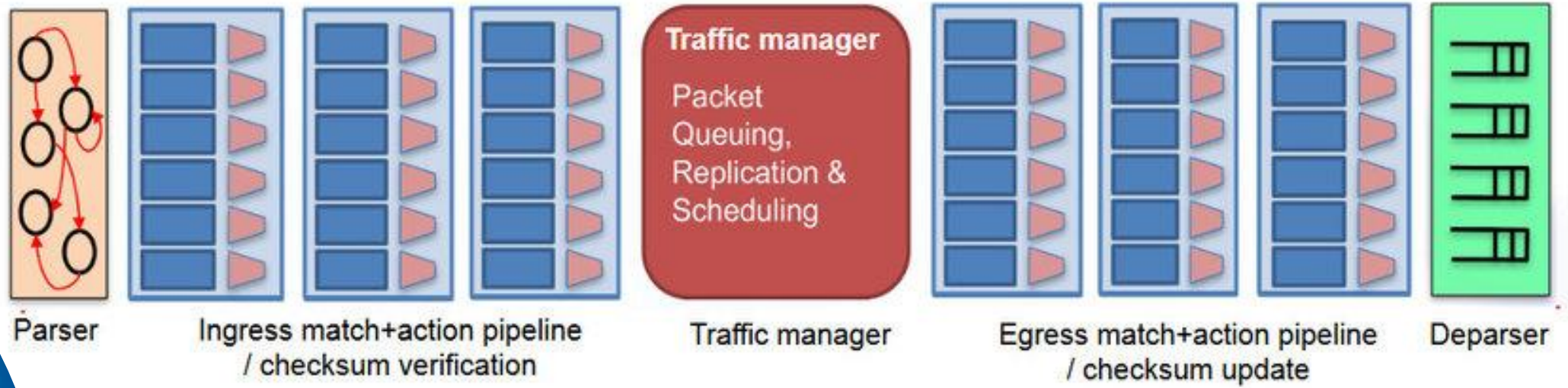
```
table l2_exact_table {
    key = {
        hdr.ethernet.dst_addr: exact;
    }
    actions = {
        set_egress_port;
        drop;
    }
    const default_action = drop;
}
```

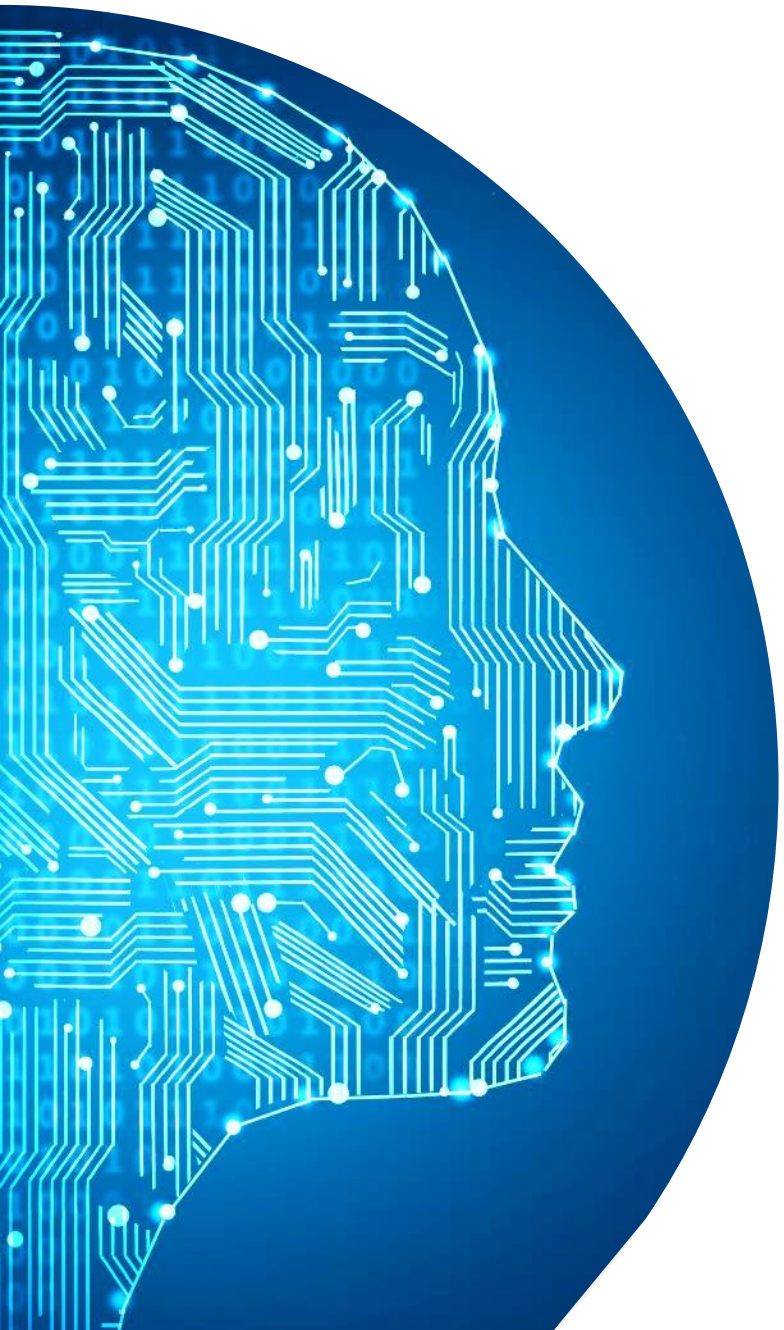
```
action drop () {
    mark_to_drop(standard_metadata);
}

action set_egress_port(port_num_t port_num) {
    standard_metadata.egress_spec = port_num;
}
```

V1 model



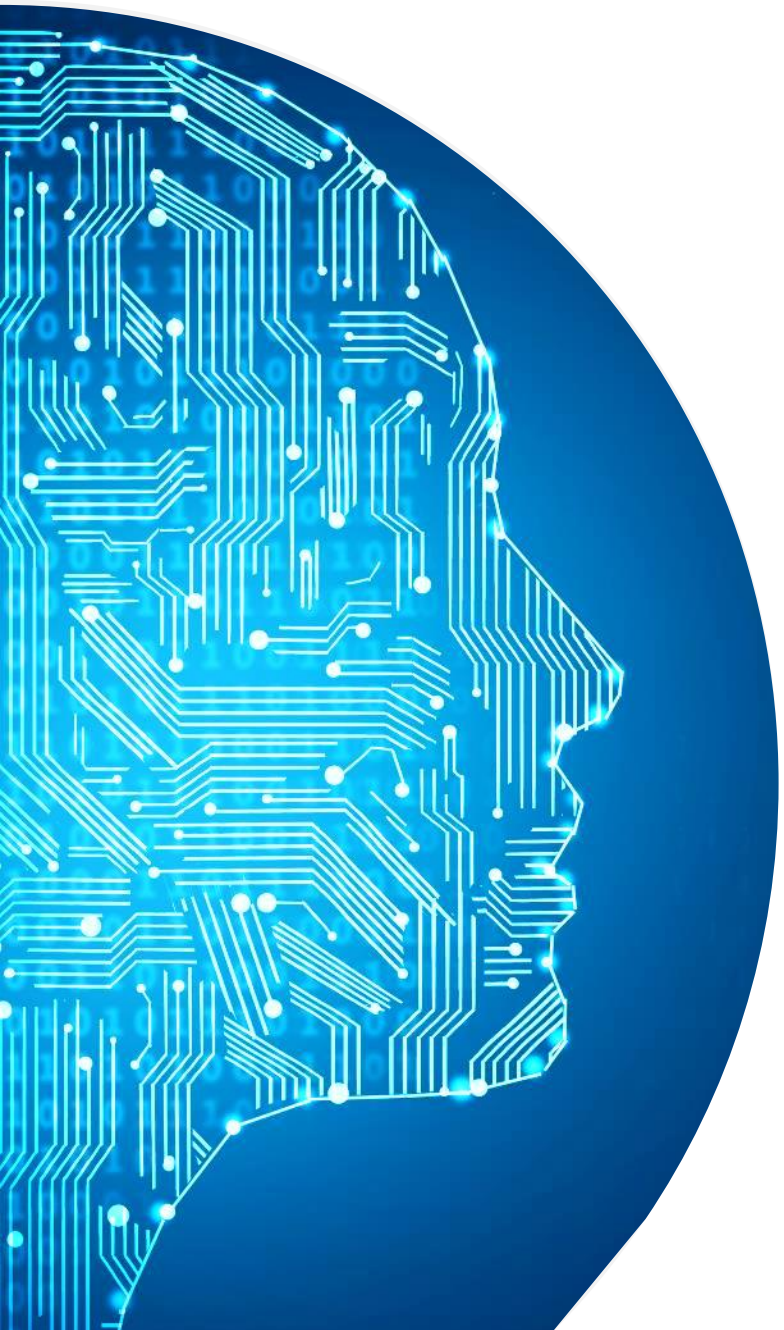




Our P4 program

- Let’s take a look at our P4 program!

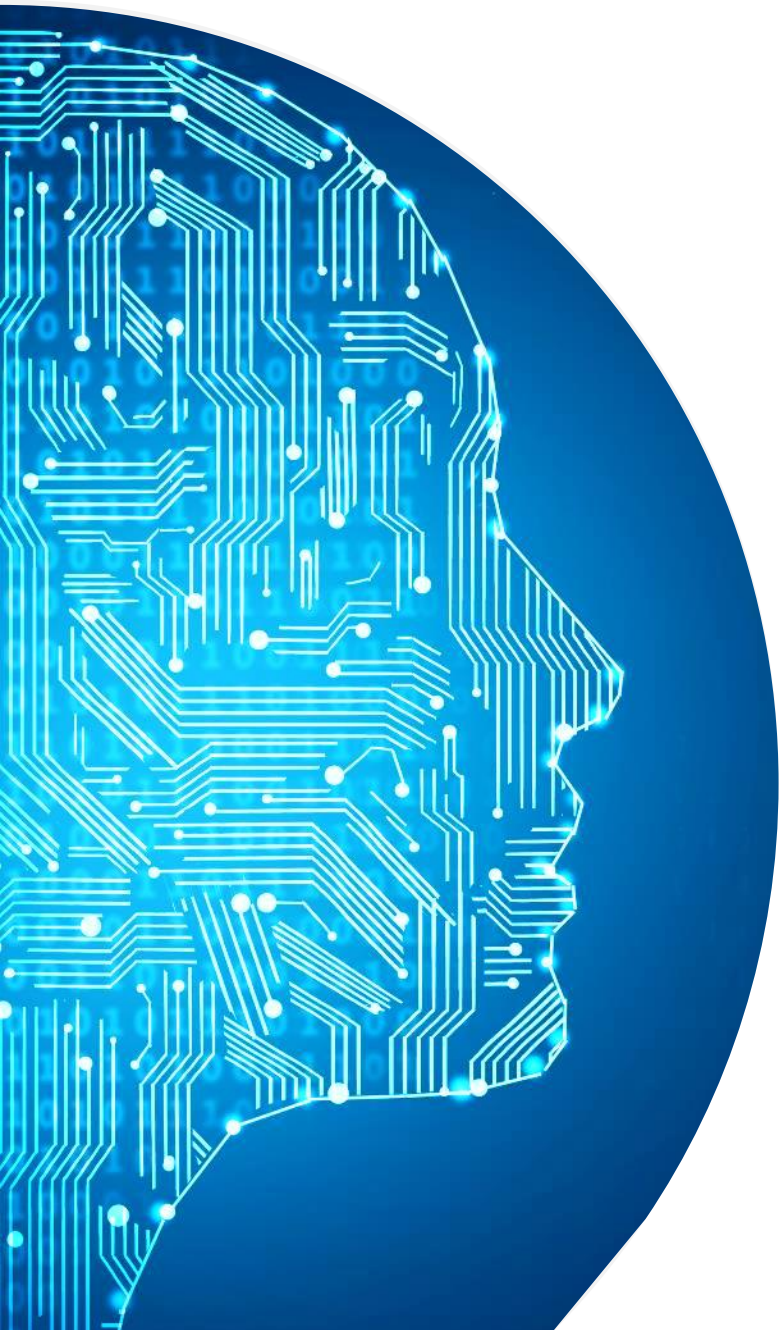
Toy case for P4 telemetry



Inband Network Telemetry (INT)

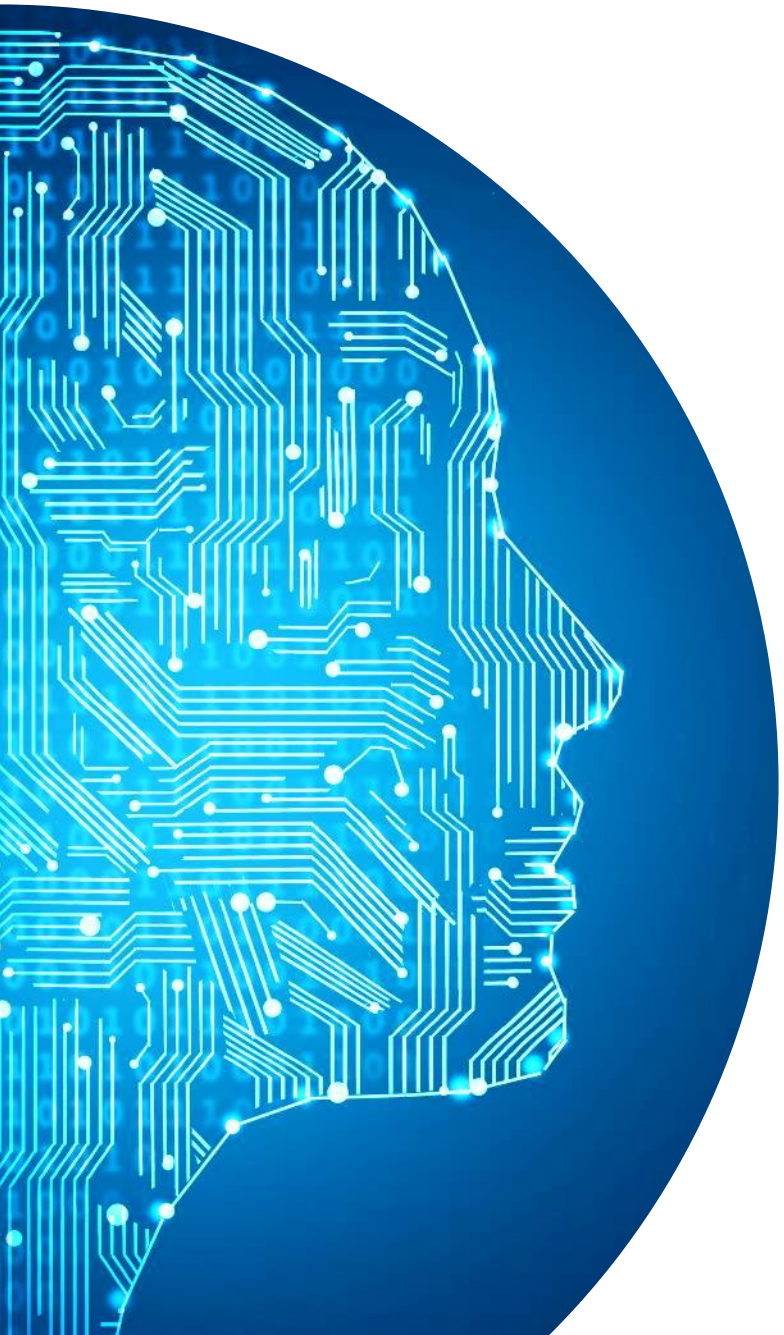
TeraFlow SDN

- The basic idea is to embed some metadata to the packets that traverse the switch
- More details will be shared in a later presentation by Georgios Katsikas
- But let’s try and see in practice how such a thing could be implemented



IPv4 options

- IPv4 headers have an optional part called O*ptions* which follows the standard IPv4 header and can be of variable length (0-40 bytes)
- IPv4 Options are (rarely) used to record routes, timestamps, security mechanisms, etc.
- We are going to use that part to embed some information from the switches

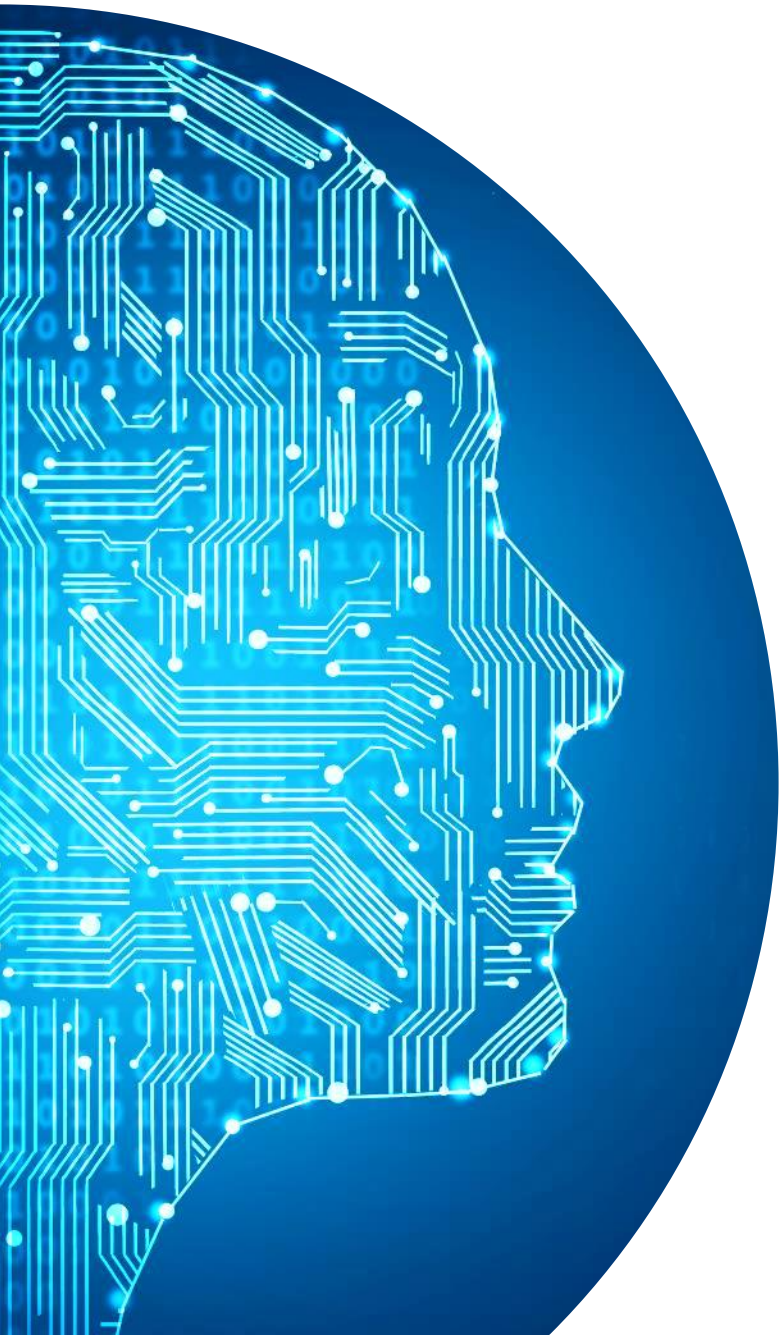


P4 Standard Metadata

- Many different things are collected by a P4 switch, including:
  - ingress_port : implementation specific
  - packet_length : implementation specific
  - egress_port : implementation specific
  - ingress_global_timestamp : 48 bits
  - egress_global_timestamp : 48 bits
  - mcast_grp : 16 bits
  - enq_timestamp : 48 bits
  - enq_qdepth : 19 bits
  - deq_timedelta : 32 bits
  - deq_qdepth : 19 bits

TeraFlow SDN

Implementation



Sender/Receiver

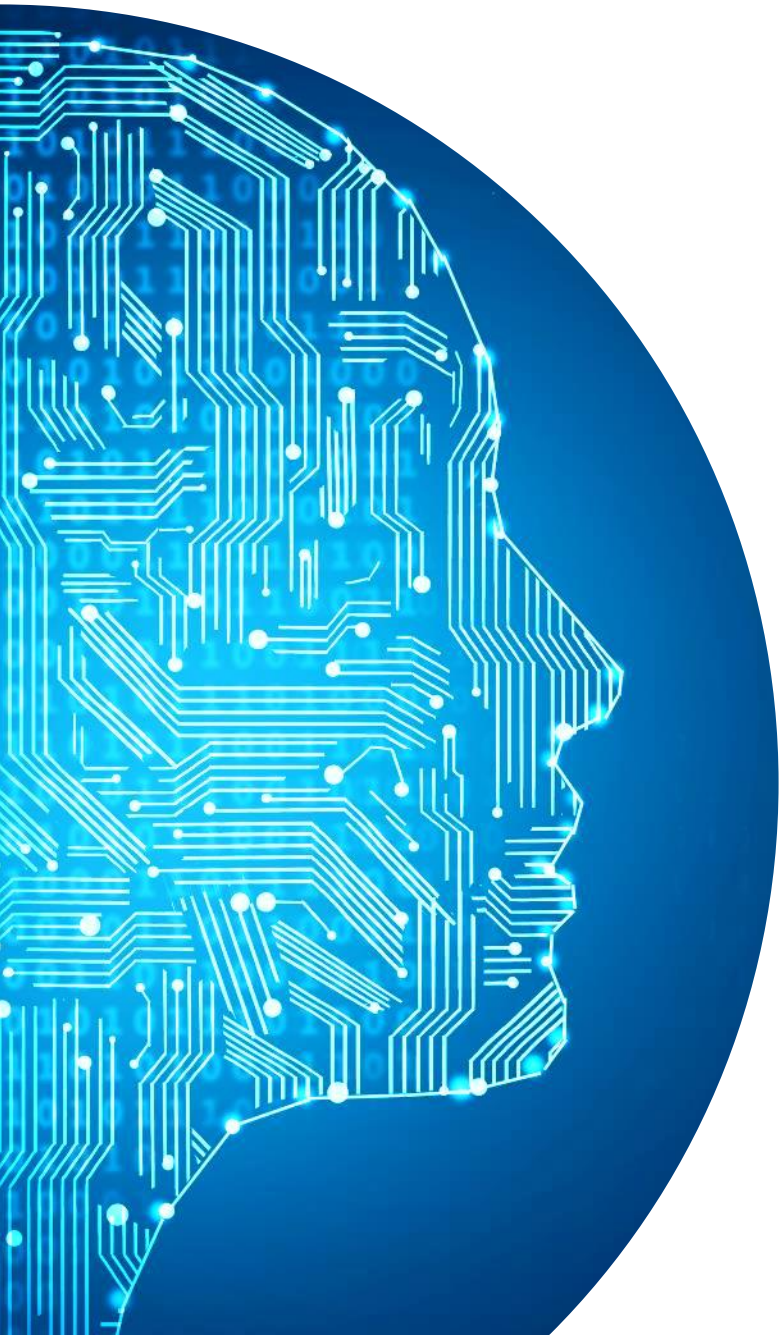
TeraFlow SDN

- First, we need a way to correctly decode our packets and extract the int information

- For that reason, we are providing two python scripts

Some extra steps are needed to run those in a Mininet container!

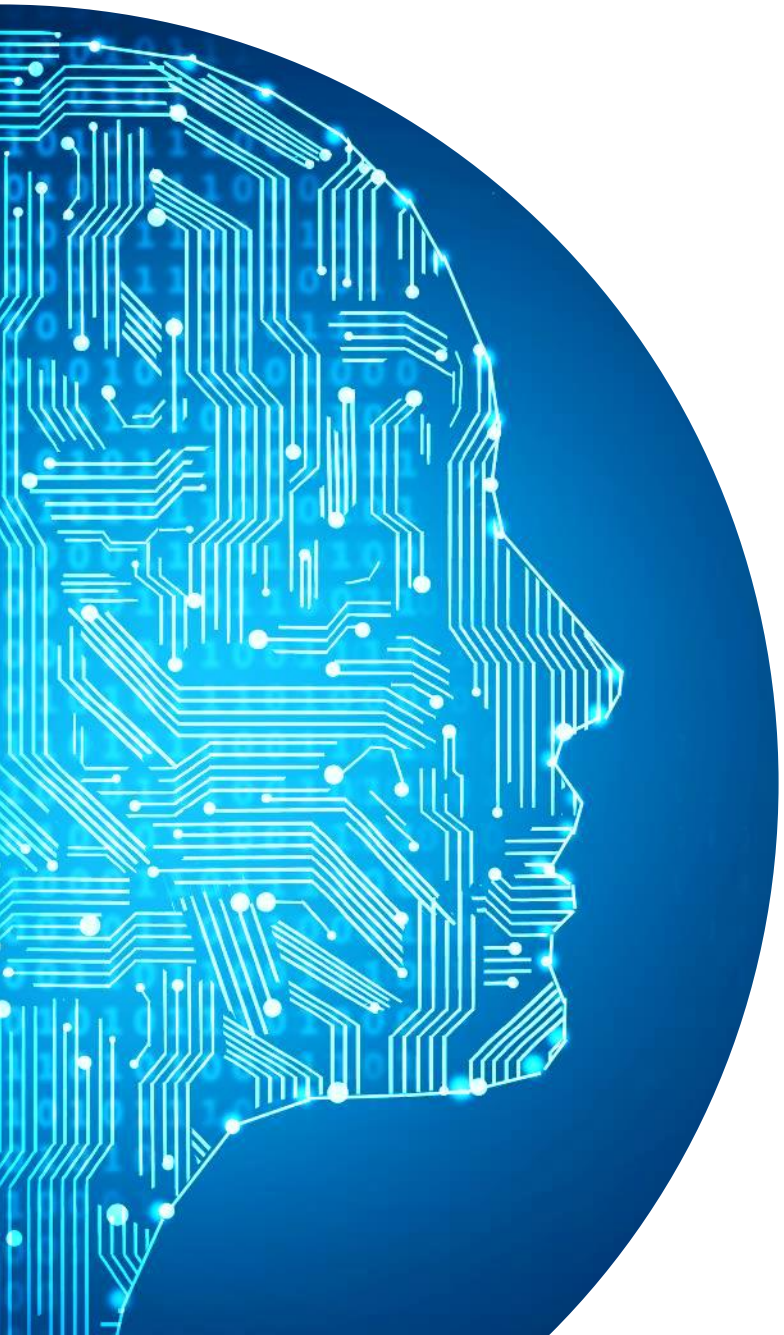
- The scripts are taken from the **Networked Systems Group (NSG)** in **ETH Zurich**
  - https://github.com/nsg-ethz/p4-learning/tree/master



P4 program

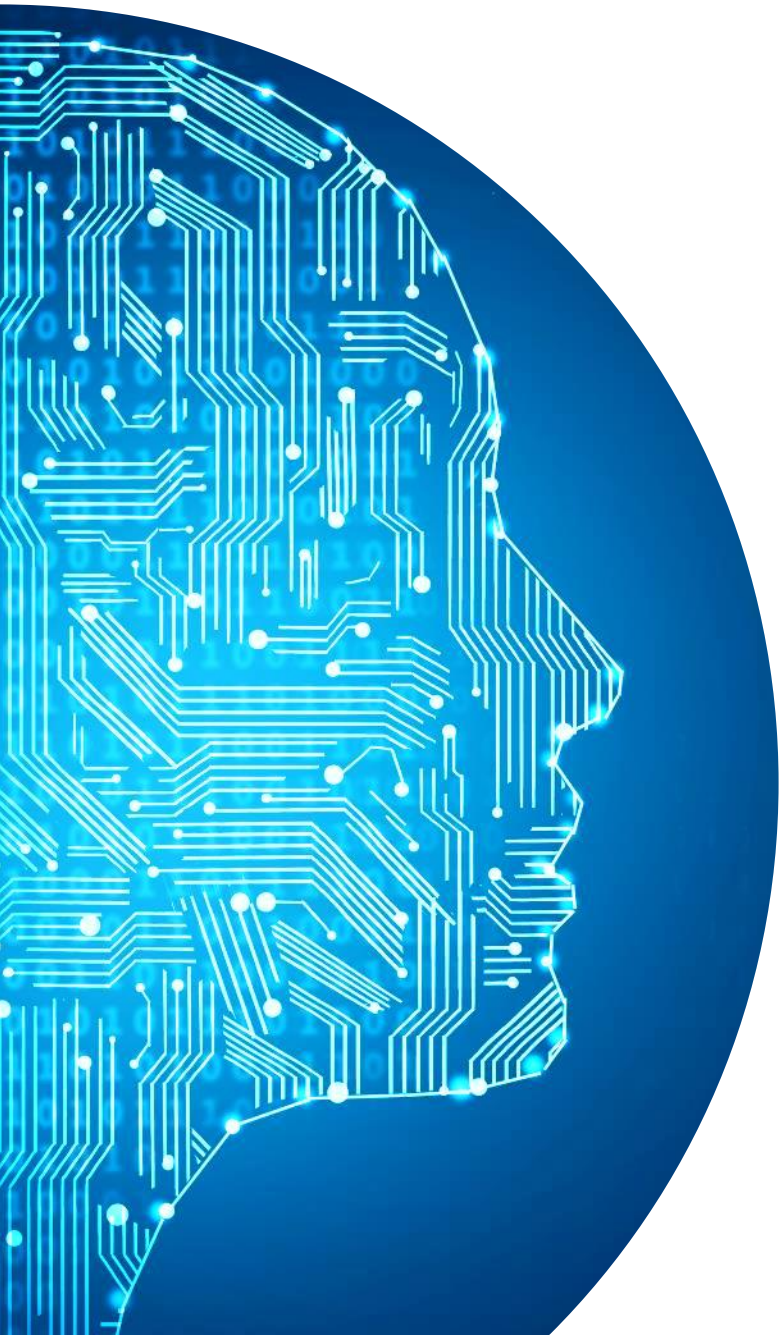
- The P4 program used encodes in each packet the following information:
  - The switch ID
  - The queue depth when the packet dequeued
  - The output port
- As it stands now, the switch ID is always the default 1
- Let’s run the experiment

It is best to redeploy TFS using SKIP_BUILD!



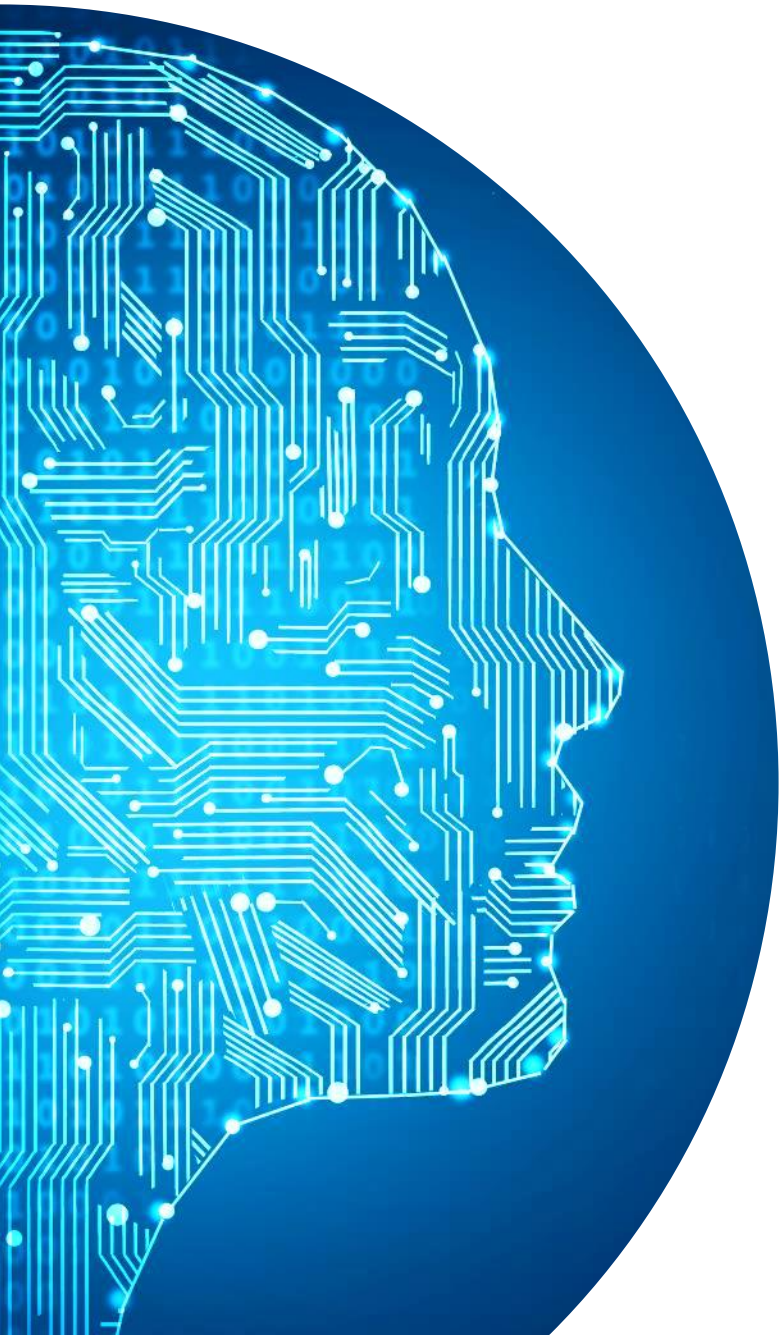
Exercise 1

- Change the P4 program to receive a custom number as switch id from the controller
- Change the Service Handler to automatically install a rule with the correct switch id



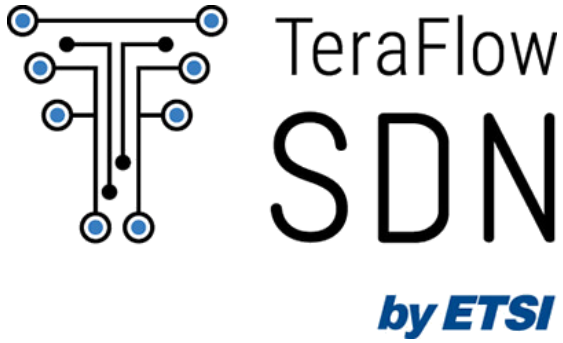
Exercise 2

- Change the P4 program and the send/receive scripts to write the timestamp of when a packet arrived on the switch
- `ingress_global_timestamp` is 48 bits, but options currently is 32 bits (maximum 40 bits)!
- For now, let's remove all the other information to accommodate timestamp



Extensions to try on your own

- Create a KPIs in Monitoring from the INT timestamp
  - Combine with the scripts we created in Session 1
- Use (or define) a new protocol, besides IPv4 Options, for INT. That will allow us more flexibility (like more than 40 bits)
  - INT specification: https://p4.org/p4-spec/docs/INT_v2_1.pdf
  - And more details following by Georgios Katsikas
- Do not hesitate to contact in slack or mail
  - pfamelis@ubitech.eu



Thank you!

TFSsupport@etsi.org