Università della Svizzera italiana

UNIVERSITY OF CAMBRIDGE

P4Debug:
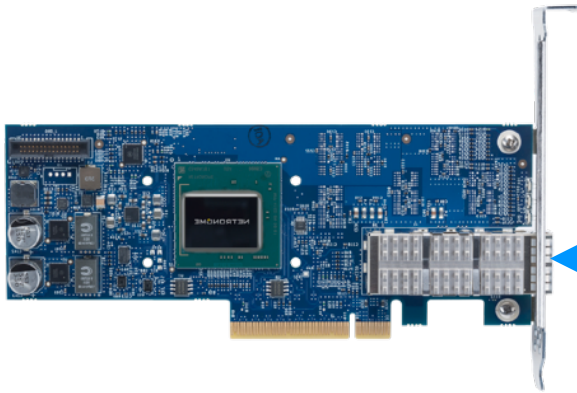
A Framework for Debugging Programmable Data Planes

Pietro Giuseppe Bressana
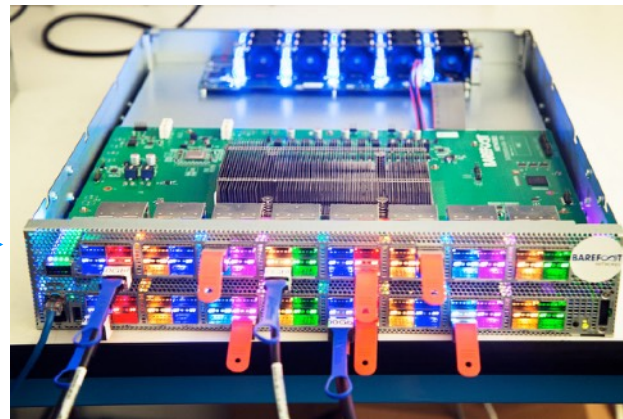
Research Advisors:
Prof. Robert Soulé
Dr. Noa Zilberman

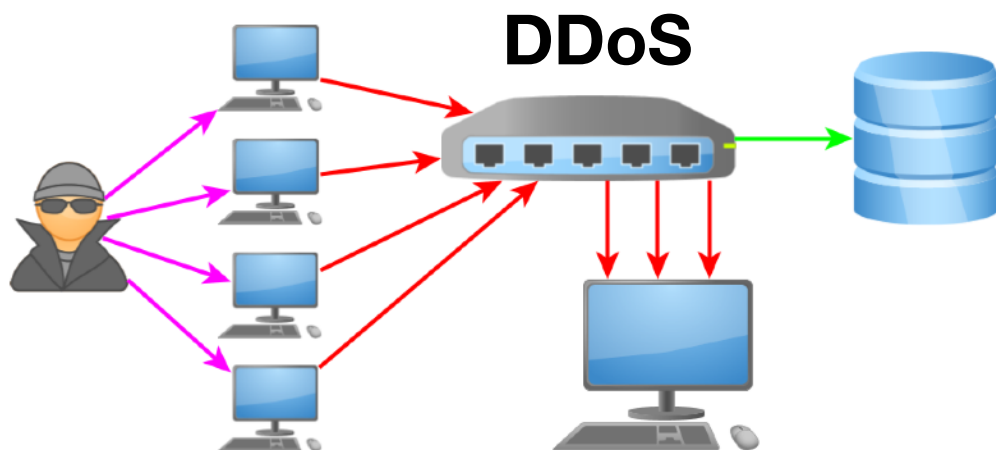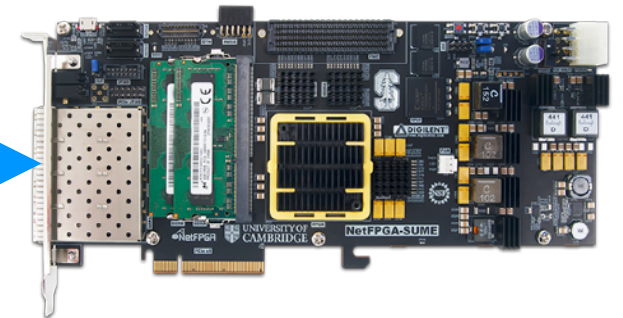# Emerging In-Network Computing

Smart NIC

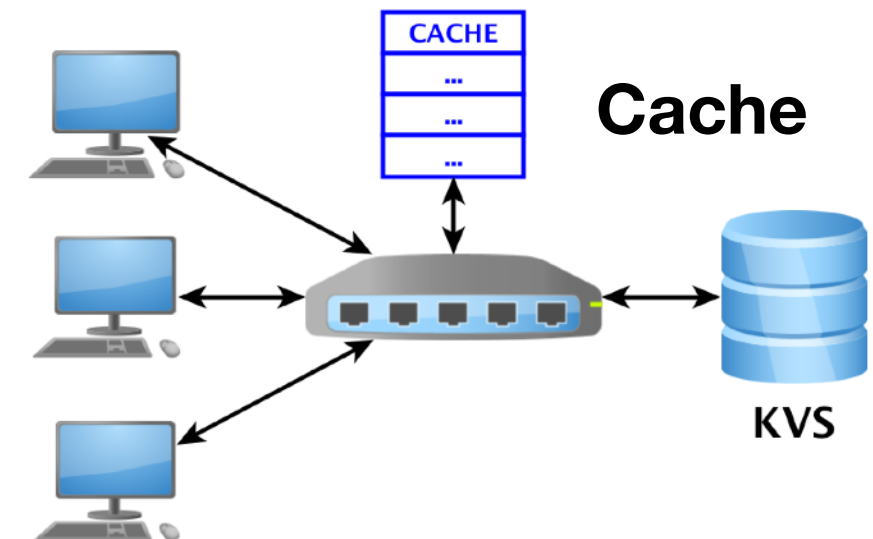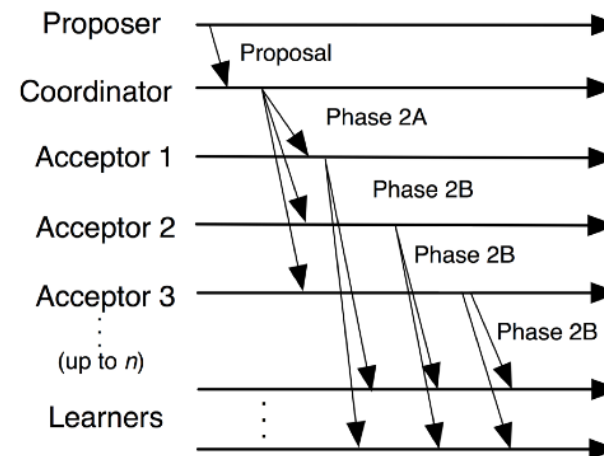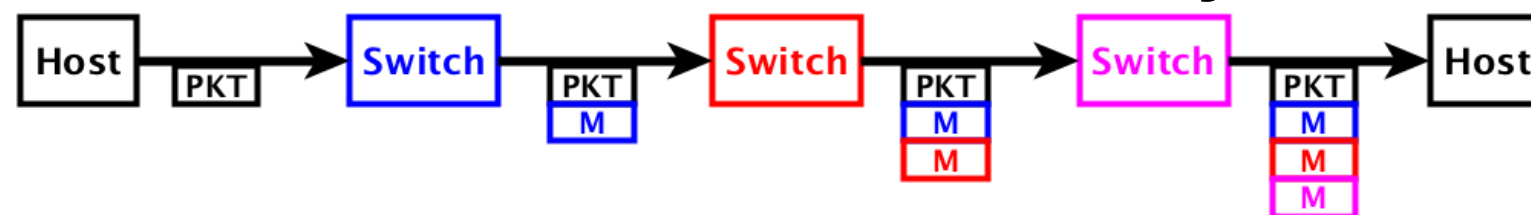**Programmable switch**

FPGA
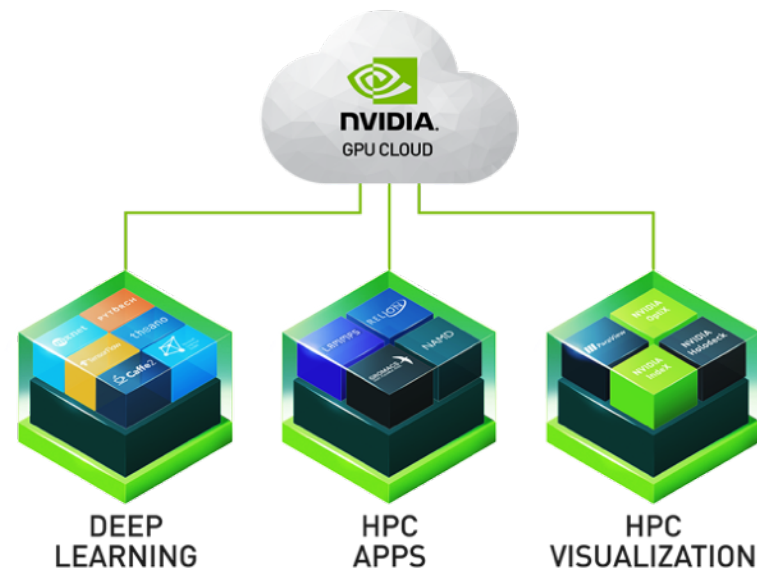

DDoS

Consensus

Cache

In-band Network Telemetry

2

Programmable Hardware In The Cloud

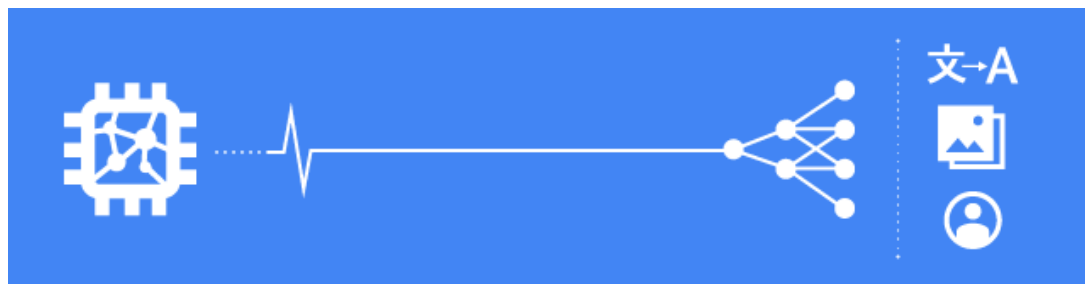**nvidia GPU CLOUD**

Microsoft Catapult
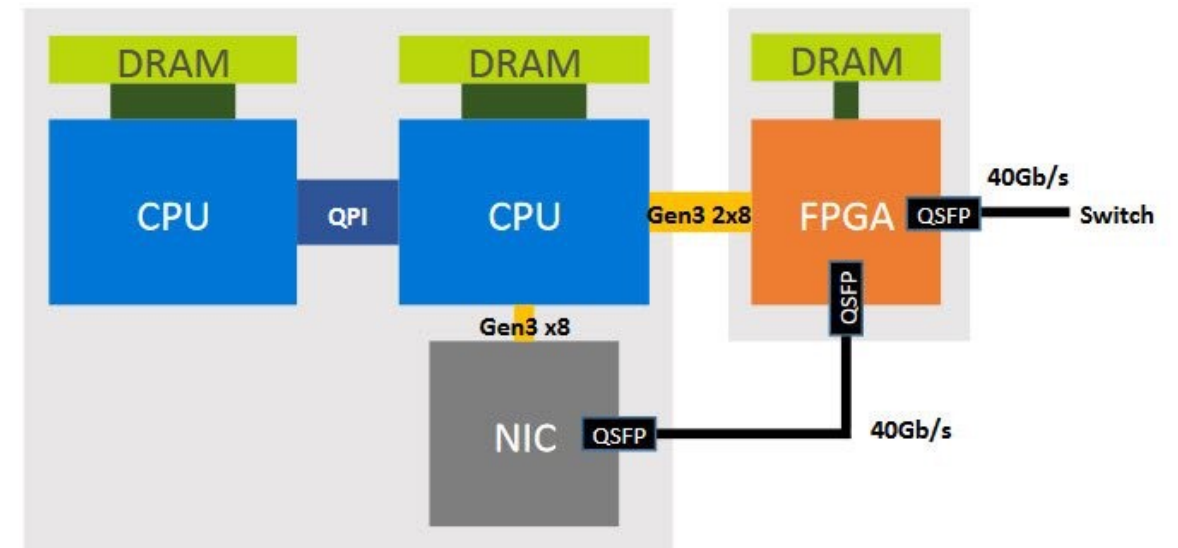
Google Cloud Platform (TPU)

Amazon EC2_F1
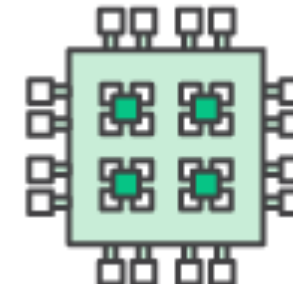
Some of these technologies are accessible to users

3

In-Network Computing: New Challenges

How to **debug** an application in **real time**?

How to **validate** (and provide full coverage) for a **deployed application**?

No software-like debugger in hardware

4
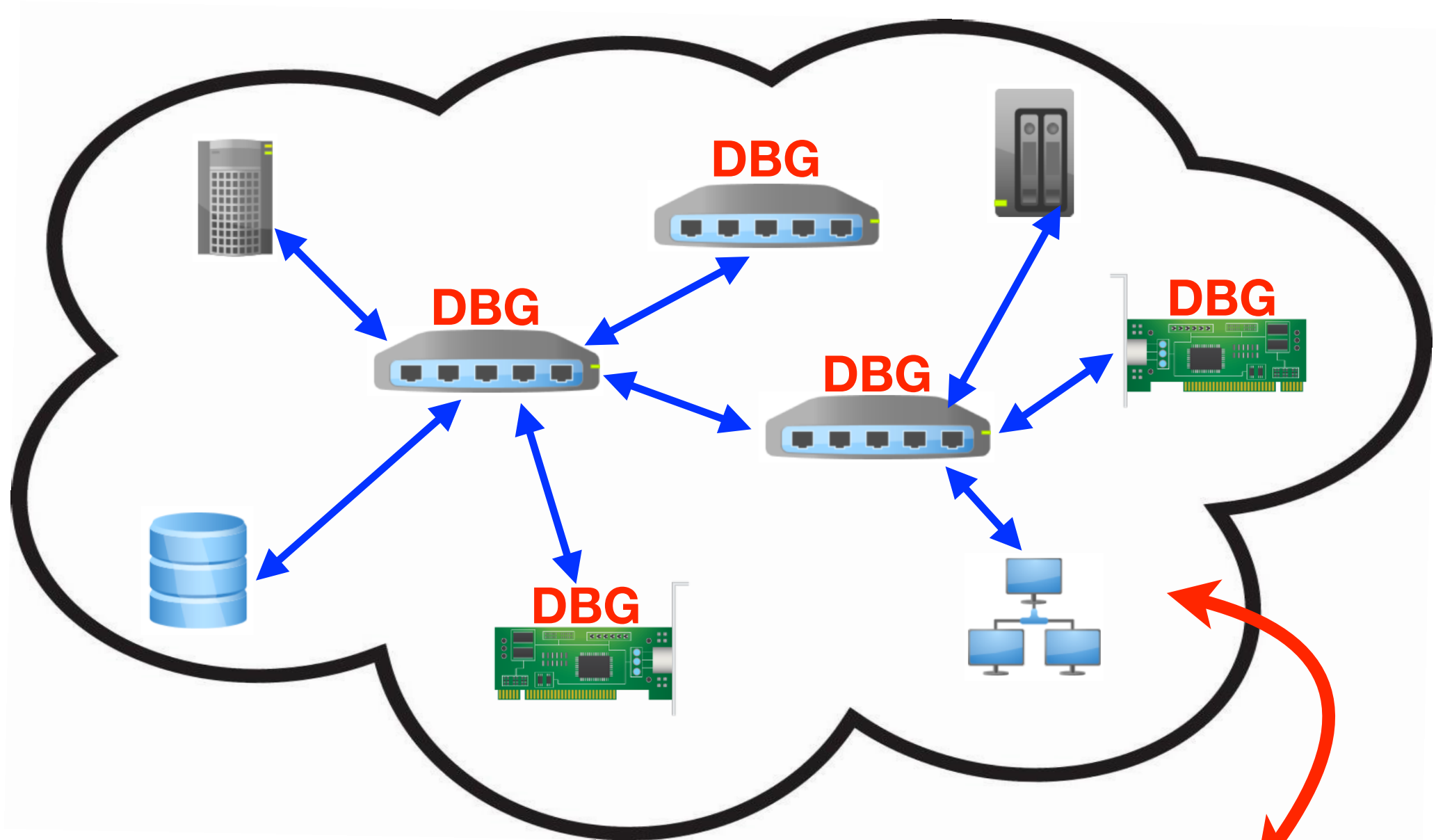
P4Debug

Validation & Real-time debugging of networked applications
(e.g. inside cloud environments)
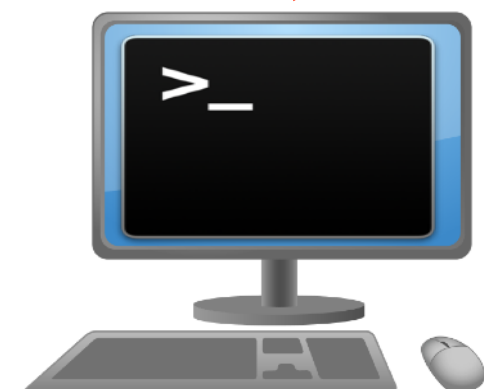
P4Debug: Programmable Architecture

Network hardware platform

TESTED DATAPLANE

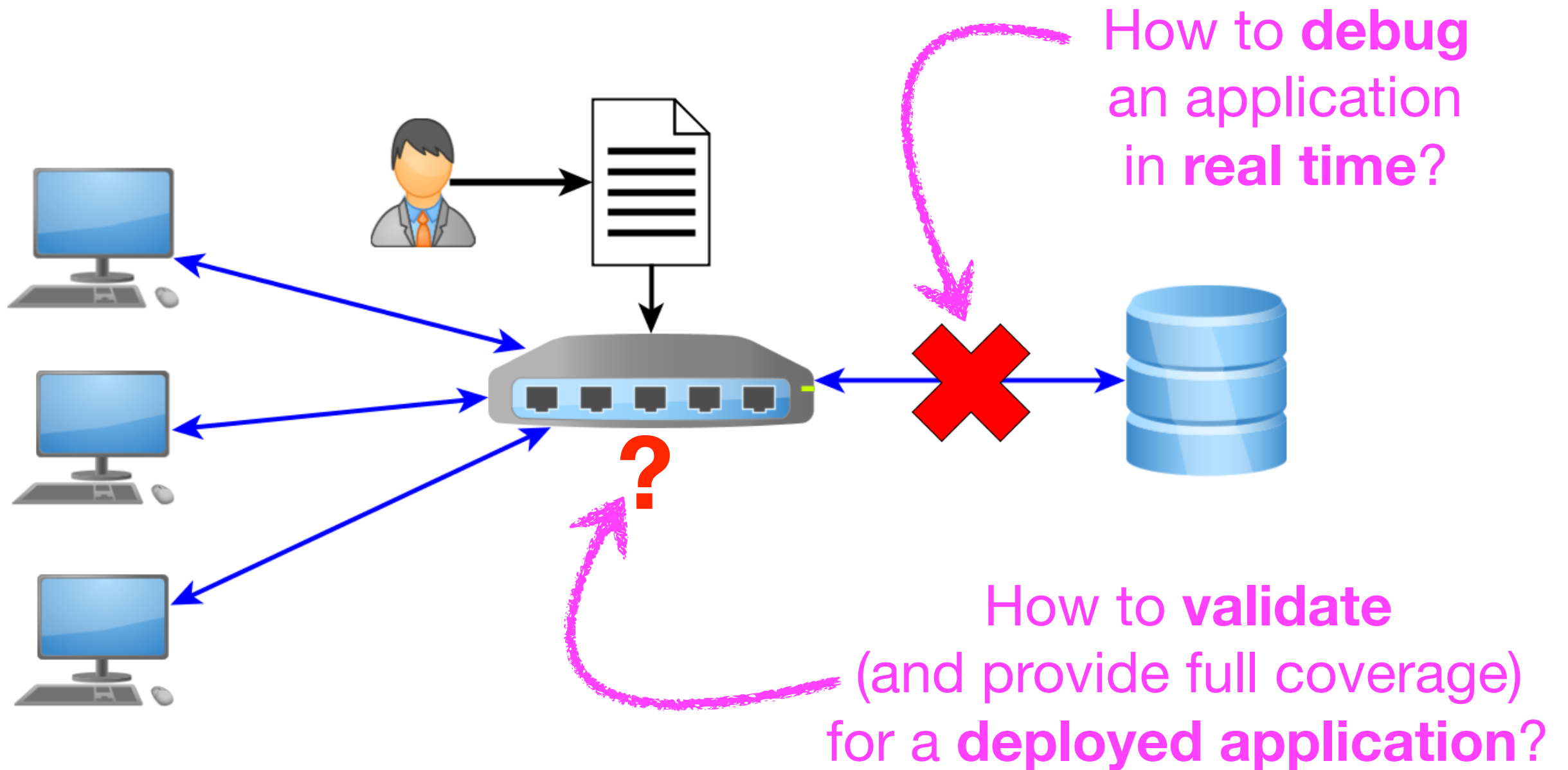INPUT INTERFACE

OUTPUT INTERFACE

DEBUG PACKET GENERATOR

VERIFIER

Programmable debug packets **generator** (inside the hardware)

Programmable **verifier** (inside the hardware)
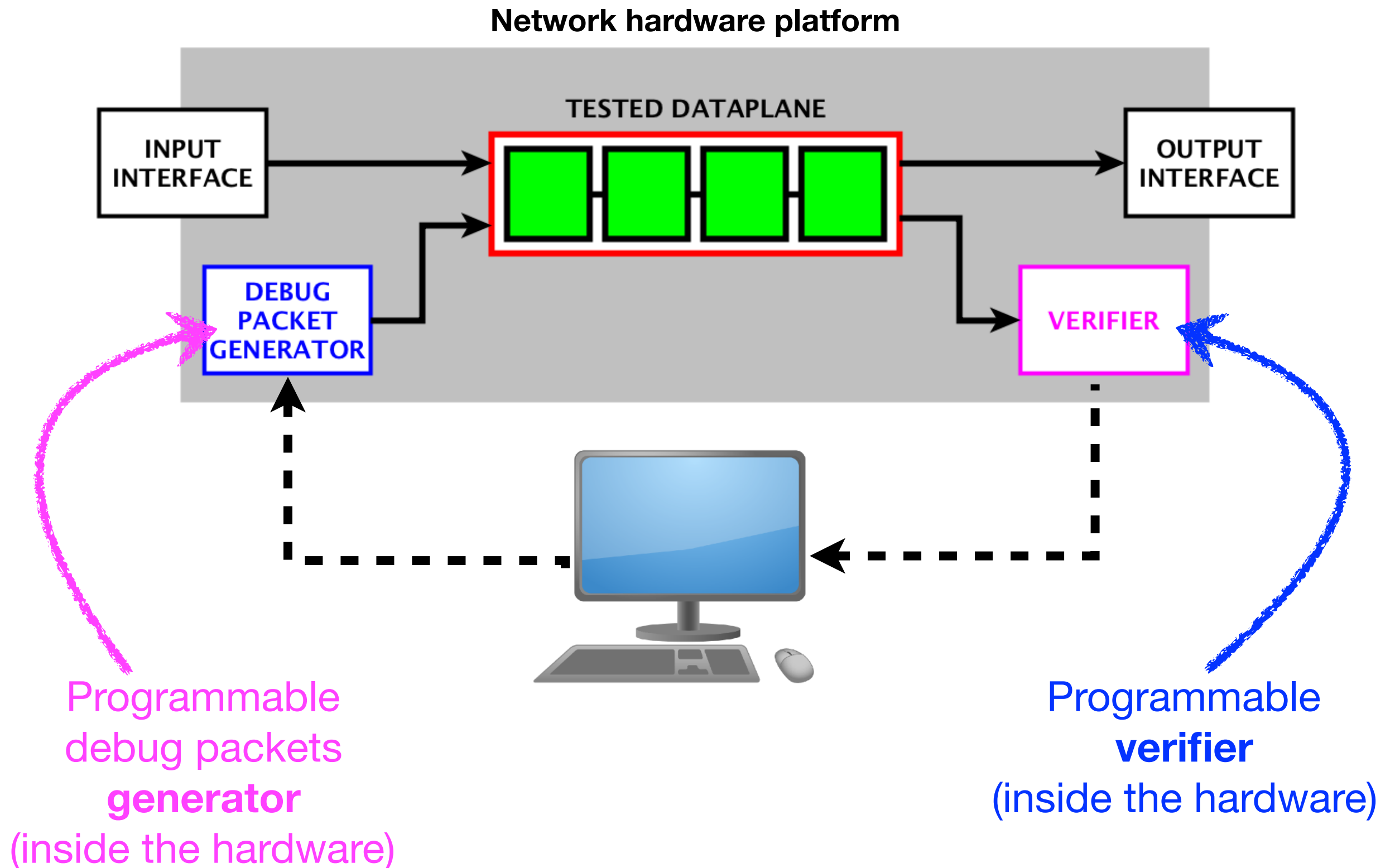
A Programmable Debugger

P4:

a language for designing programmable data planes

P4Debug:

create programmable application-specific debugging environments

Using Formal Verification Tools

Program:
do something,
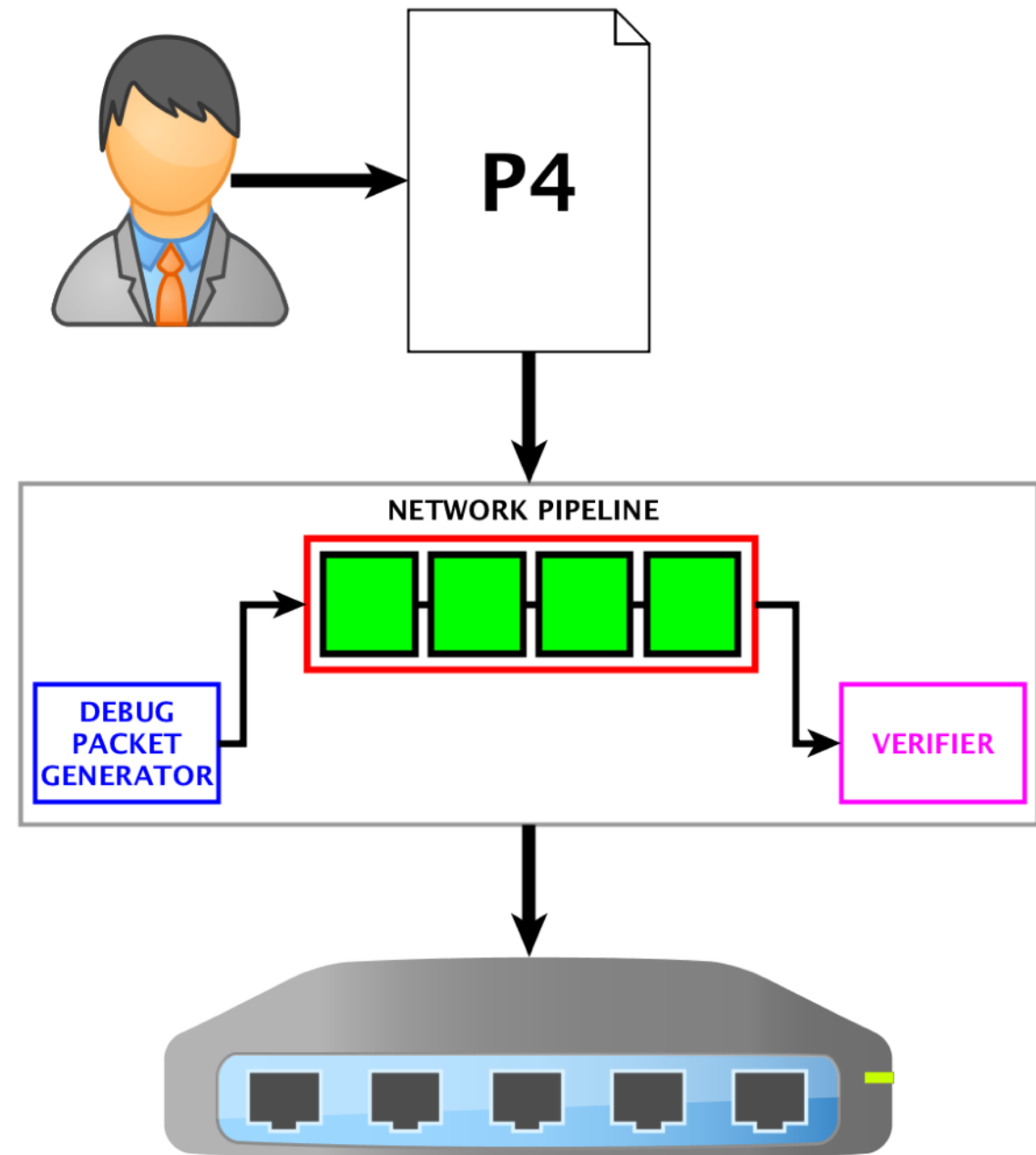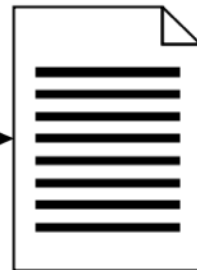use specific
protocol

Parser code:
accepts specific headers,
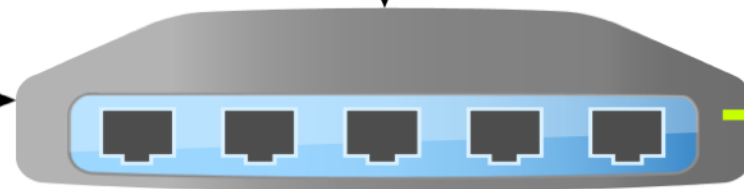**rejects others**

FORMAL VERIFICATION TOOL ✔ Verification p**assed**

PACKET

PACKET
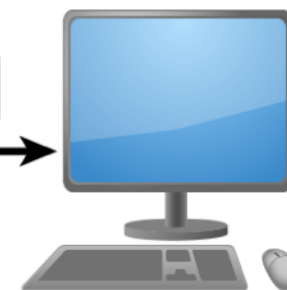
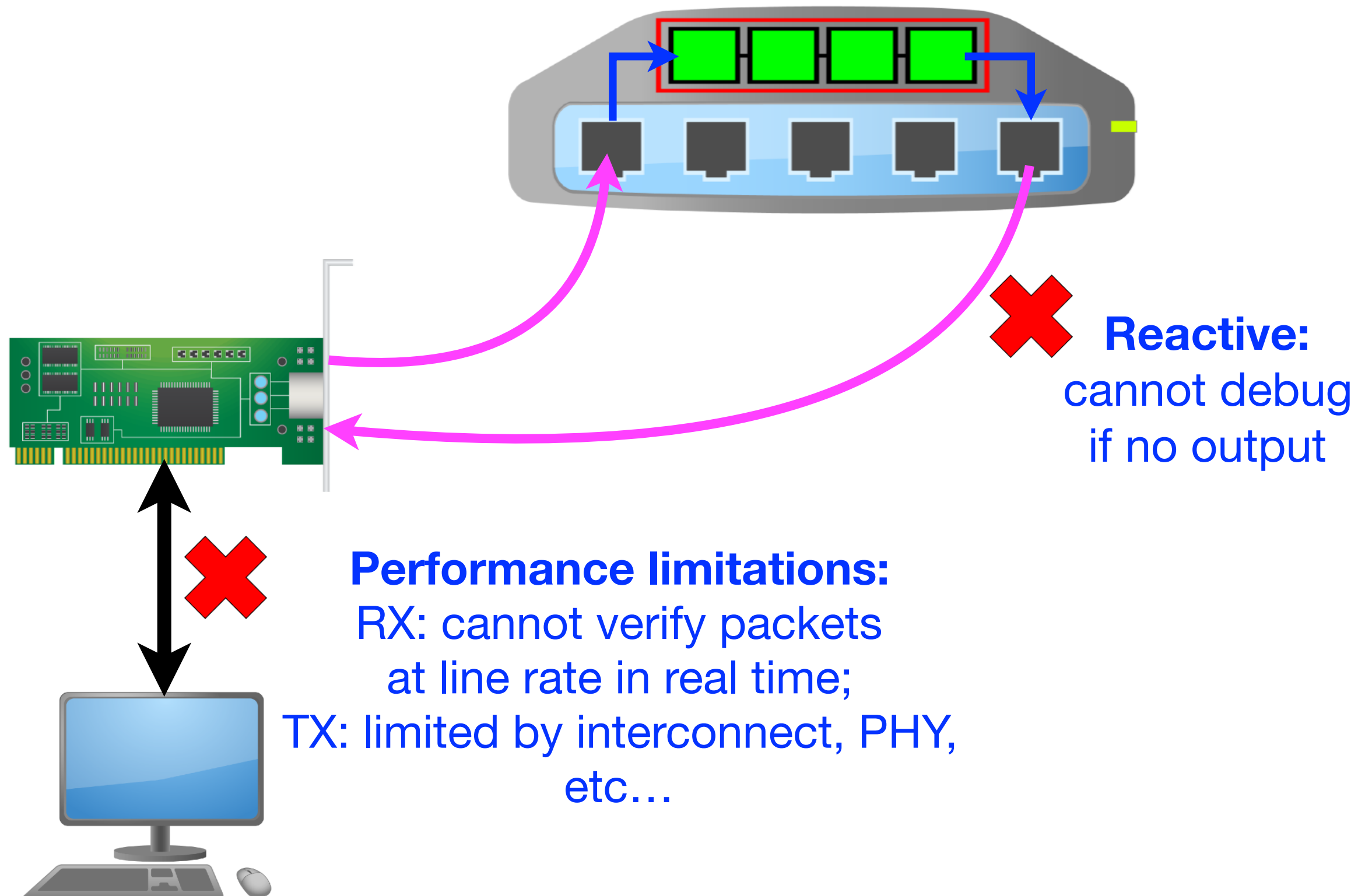**All packets pass!!!**
Nothing rejected
(dropped)

Reject function **not implemented**
by the compiler!
&
no warning provided!

Some bugs may be related to
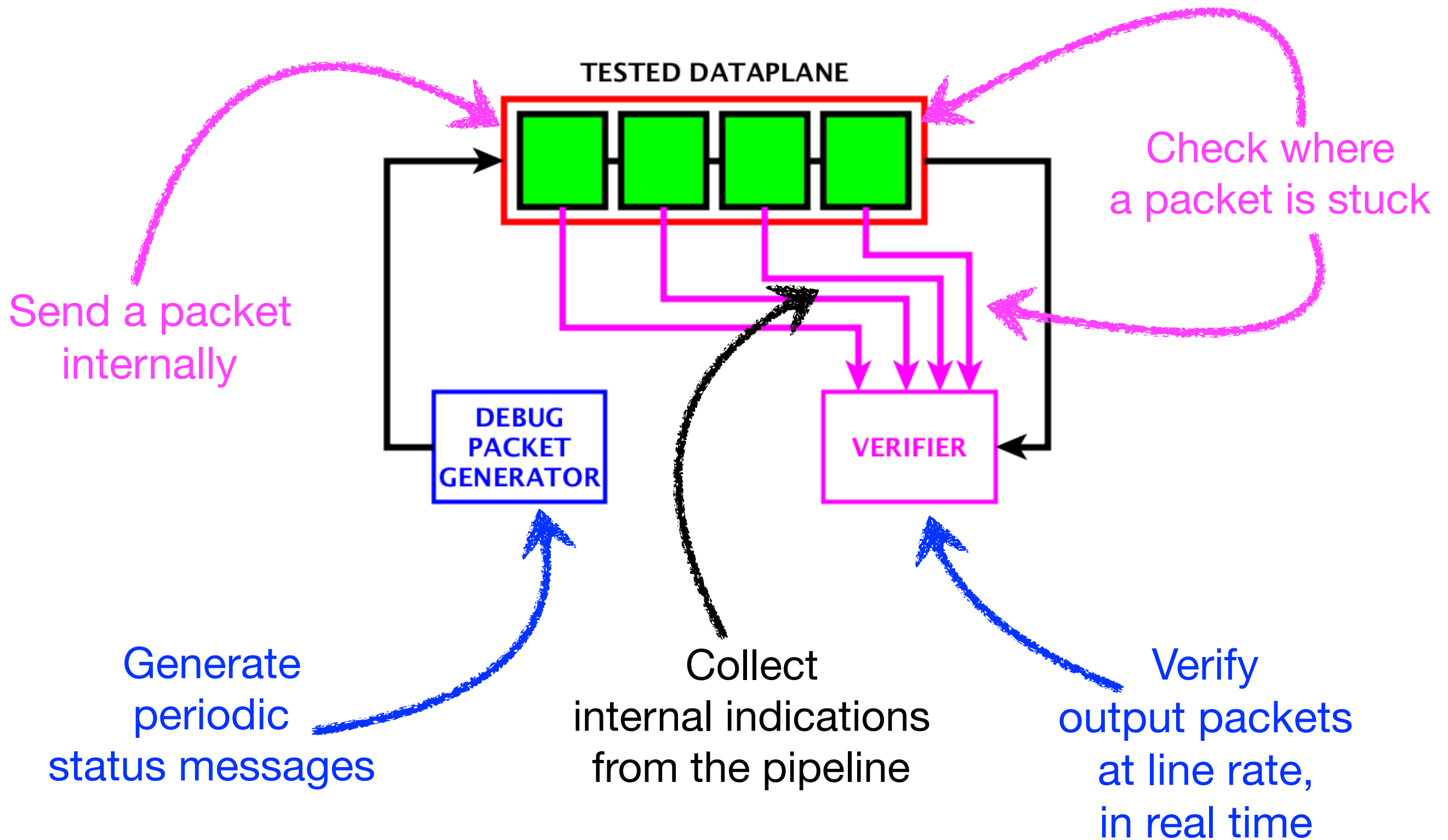the **compiler**
&
to the **architecture**

Using External Debuggers

Reactive:
cannot debug
if no output

Performance limitations:
RX: cannot verify packets
at line rate in real time;
TX: limited by interconnect, PHY,
etc…

Debug Information

TESTED DATAPLANE

Check where
a packet is stuck

Send a packet
internally

DEBUG
PACKET
GENERATOR

VERIFIER

Generate
periodic
status messages

Collect
internal indications
from the pipeline

Verify
output packets
at line rate,
in real time

10
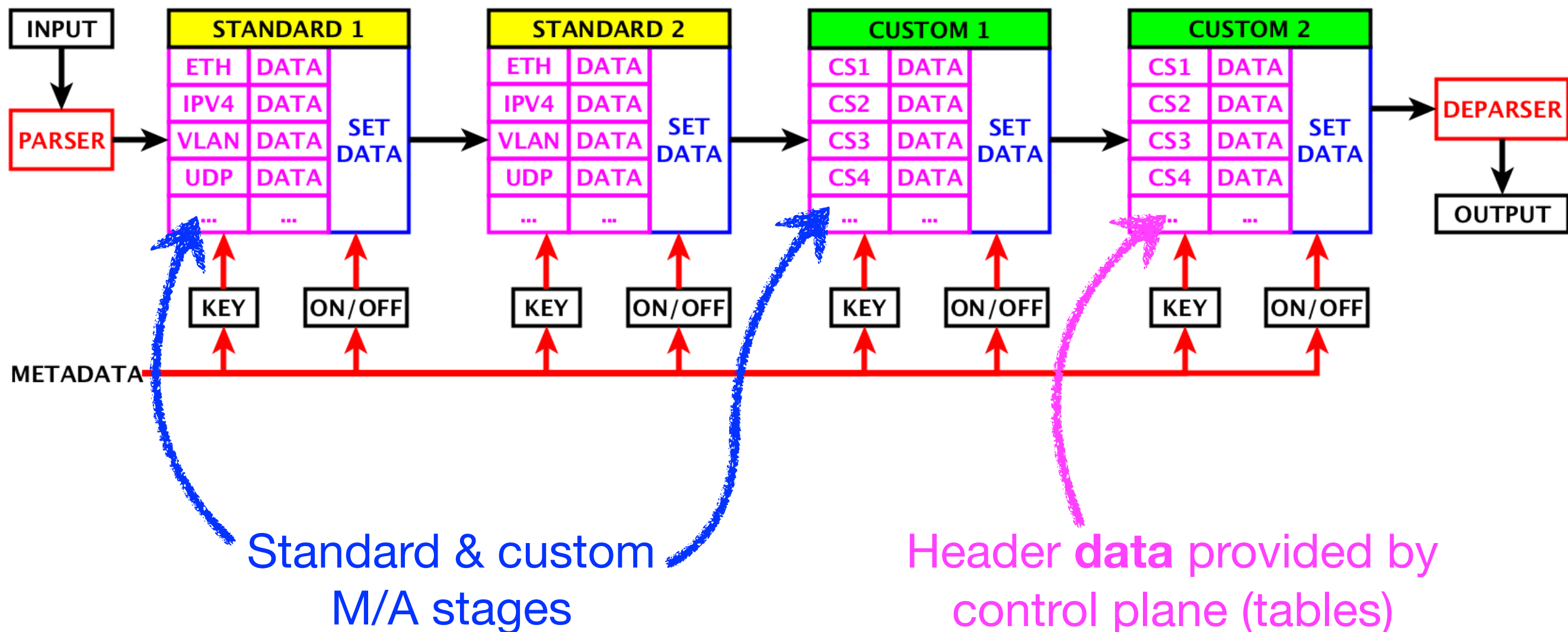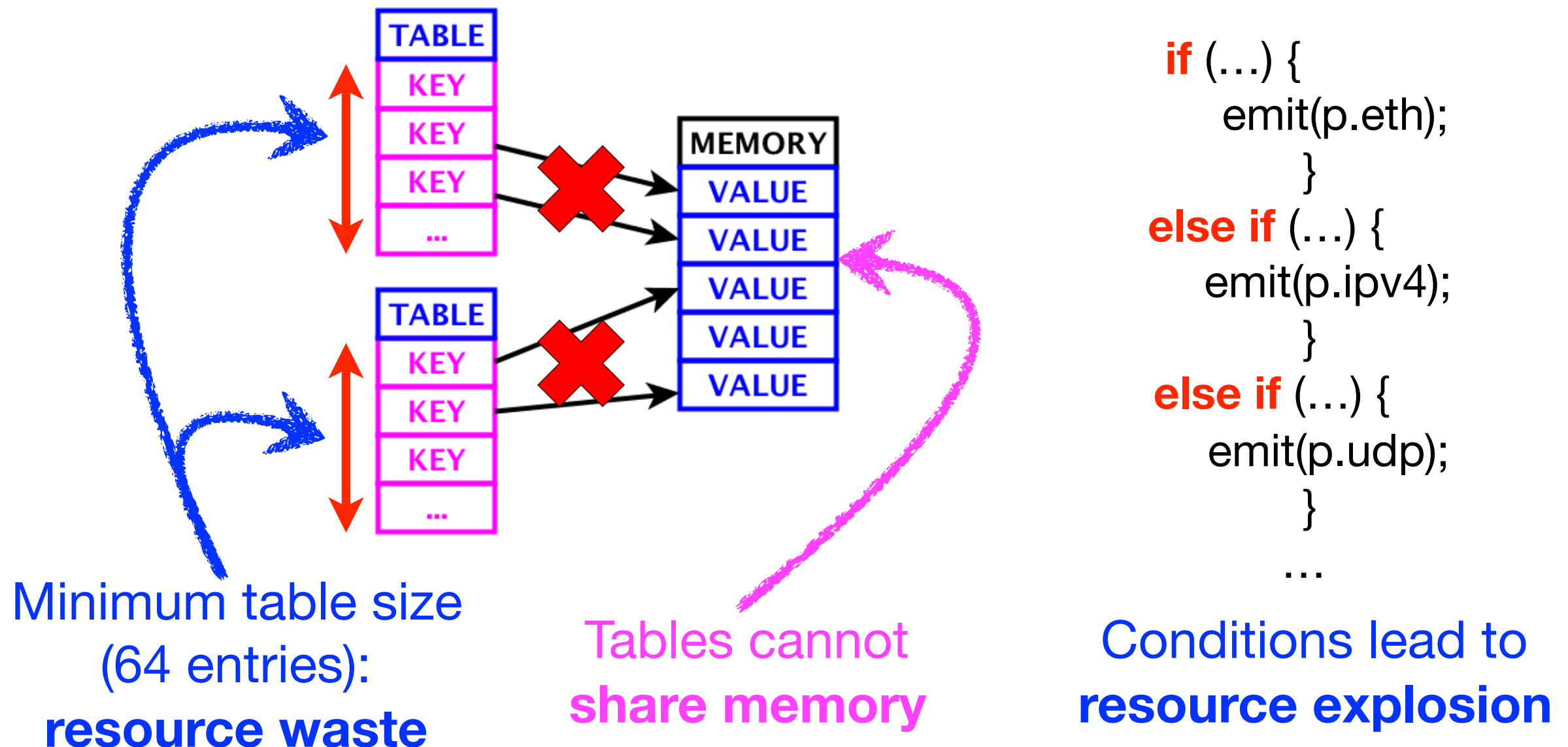
The Flexible Pipeline …

Prototype debug
packet **generator**

Can test many programs
without recompiling
the design
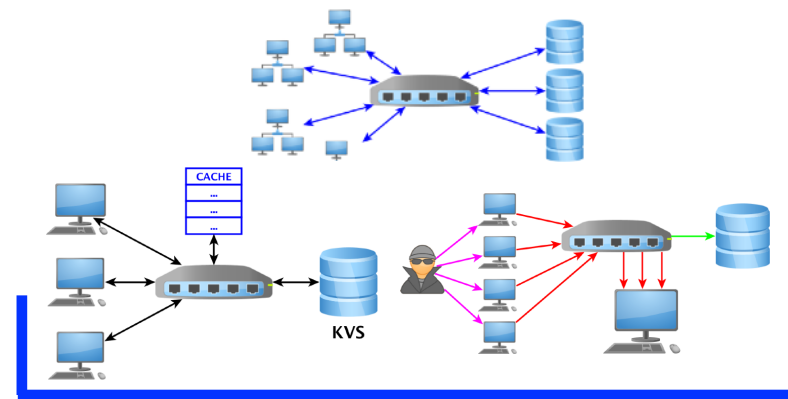
Generate sequences
of **customisable**
packet headers

Standard & custom
M/A stages

Header **data** provided by
control plane (tables)

… Killed By Closed-Source Compilers

No open-source P4 to hardware compiler,
closed-source compilers are **limited**:


```
if (…) {
    emit(p.eth);
        }
else if (…) {
    emit(p.ipv4);
        }
else if (…) {
    emit(p.udp);
        }
        …
```

Minimum table size
(64 entries):
resource waste

Tables cannot
share memory

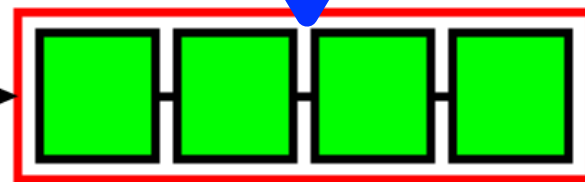Conditions lead to
resource explosion

Need to balance flexibility & resource usage

Evaluation of P4Debug

Flexibility:
number of use-cases tested
with the same P4Debug
configuration

Functionality:
coverage of possible
combinations,
number of bugs found, …

Resources:
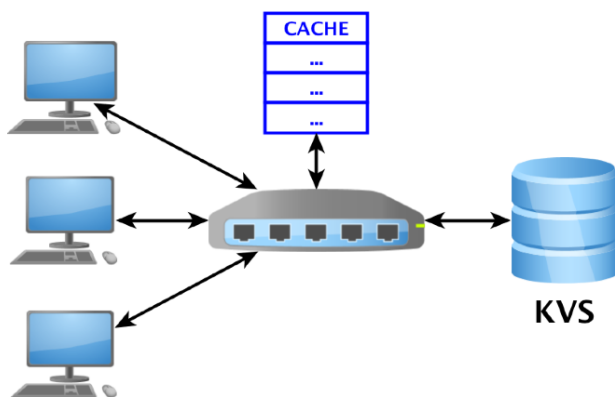overhead of adding
P4Debug

Performance:
overhead of adding P4Debug,
max rate that can be debugged,
runtime, …

13

Ongoing & Future Work
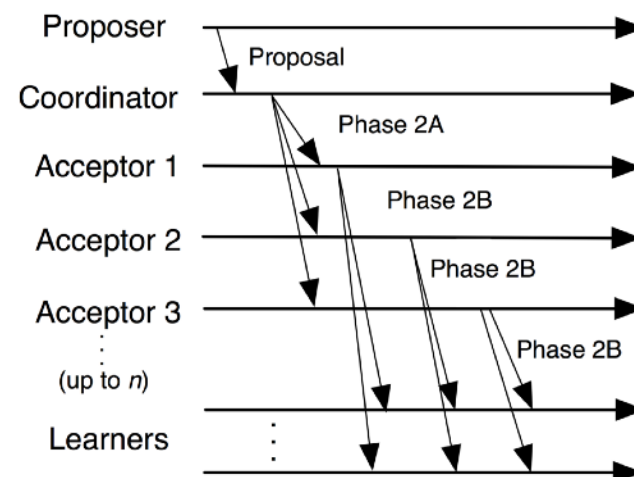
✓ Programmable generator;

✓ Programmable verifier;

✓ Management interface;
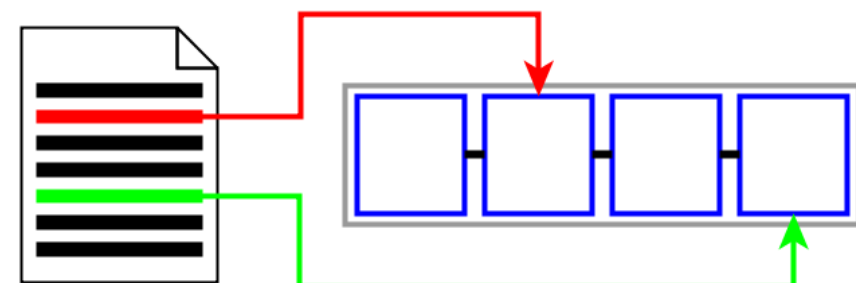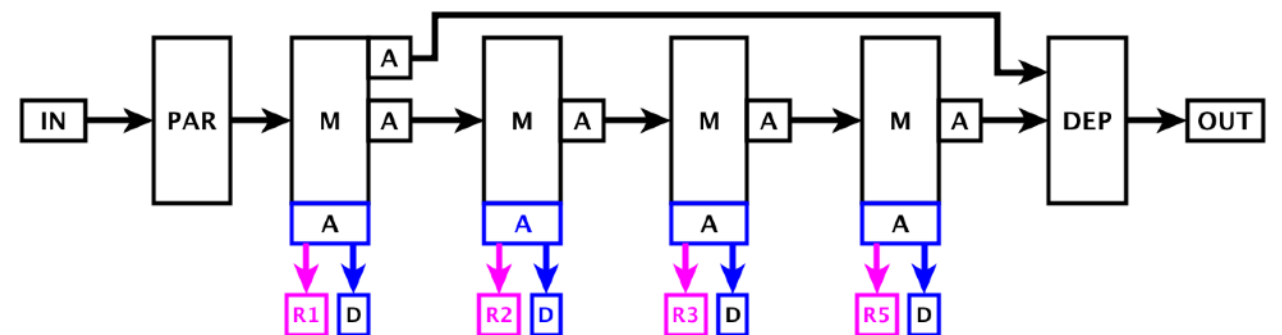
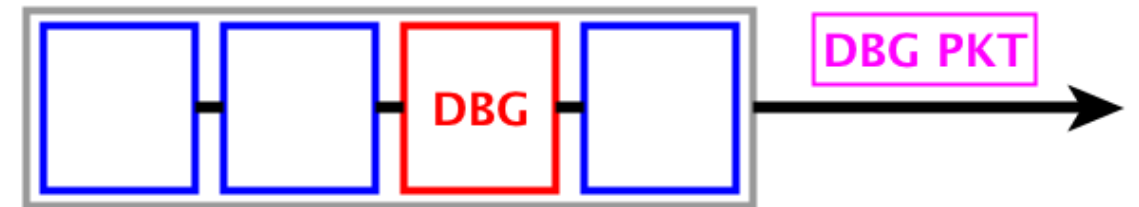✓ Tested P4 learning switch;

➡ More use-cases:
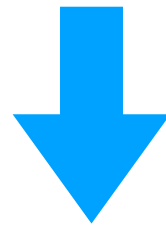
Network caching

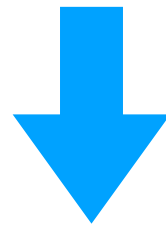**Consensus**

Language extensions:

Conclusions

Emerging **in-network computing**

Need a **validation** & **real-time debugging** infrastructure

Programmable data-planes: great **flexibility**

P4Debug: debugging network applications

enables validation

detects bugs hidden from verification tools

programmable & open-source

Summary & Questions

pietrobressana.github.io - pietro.bressana@usi.ch

16