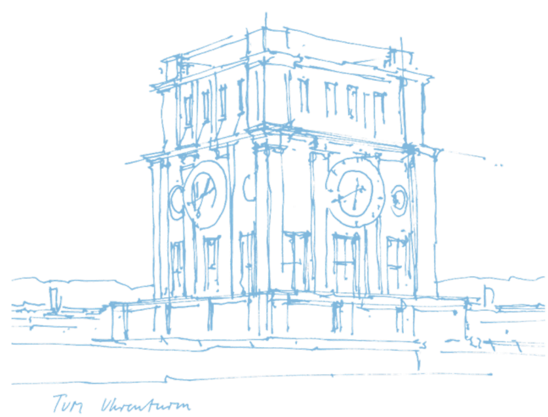
Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Rapid Prototyping of Avionic Applications Using P4

Dominik Scholz, Fabien Geyer, Sebastian Gallenmüller, Georg Carle

german-french academy
for the industry of the future
founded by IMT and TUM



Motivation

What is this talk about?

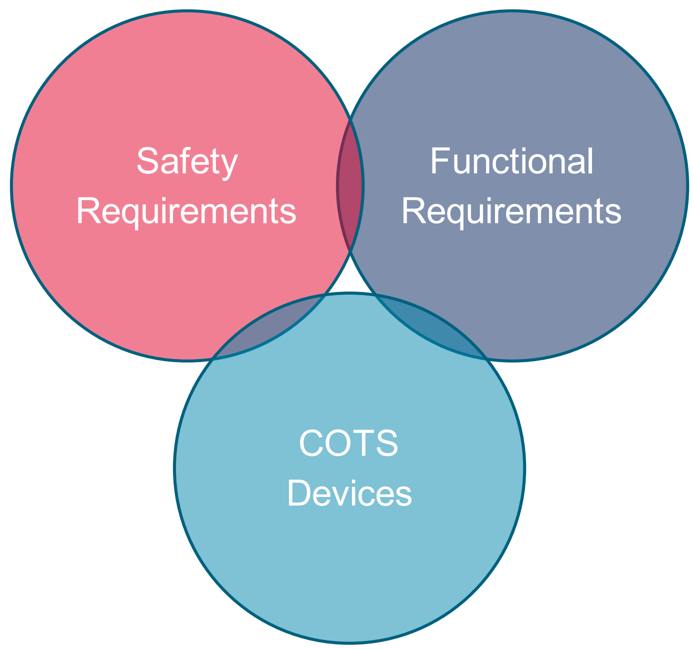
Image from https://bit.ly/2LHVmDZ

You will see. . .

- . . . P4 applied to a domain P4 was not specifically designed for . . .
- . . . what works . . .
- . . . what performance can be achieved . . .
- . . . and what is not explicitely supported by the P4 language

Motivation

The Challenge



Aircraft networks of the future

- Cheap (COTS) devices
- Implementing all required functions
- Not implementing anything else (safety!)
- . . . and short development times!

→ Ethernet-based networks ✓
→ P4?

Background: Reproducible Performance Measurements

TUM Testbed for Automated Network Experiments

Automated network experiments [1]

- Physical hosts
- Different wired network setups, also VM setups
- MoonGen [2] packet generator

Integrated P4 targets

- P4@ELTE/t4p4s
- PISCES
- p4c-behavioral/bmv2
- NetFPGA SUME
- Netronome Agilio Smart NIC

[1] S. Gallenmüller, D. Scholz, F. Wohlfart, Q. Scheitle, P. Emmerich, G. Carle, “High-Performance Packet Processing and Measurements” in 10th International Conference on Communication Systems & Networks (COMSNETS 2018), Bangalore, India, Jan. 2018

[2] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, G. Carle, “MoonGen: A Scriptable High-Speed Packet Generator,” in Internet Measurement Conference (IMC) 2015, Tokyo, Japan, Oct. 2015.

Outline

Avionics Full-Duplex Switched Ethernet (AFDX)

AFDX Implementation with $P4_{14}$

P4 Enhancements for Avionics Use Cases

Conclusion

Avionics Full-Duplex Switched Ethernet (AFDX)

Properties of AFDX

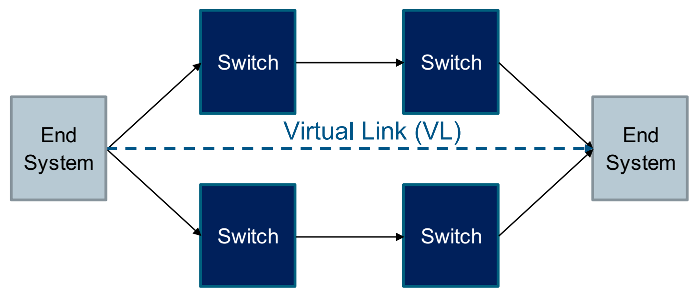
- Based on Ethernet
- Guaranteed bandwidth
- Guaranteed end-to-end latency
- Quality of Service guarantees
- Redundancy

Avionics Full-Duplex Switched Ethernet (AFDX)

Terminology

Virtual Link

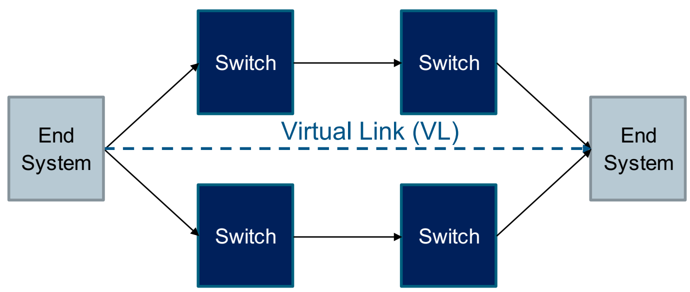
- Single source, multiple destinations
- Rate-constrained network tunnel
- Dedicated bandwidth allocation
- Static configuration



AFDX Implementation with $P4_{14}$

Basic Components of AFDX Switch

1. Frame format
2. Integrity Checks
3. Static Forwarding per Virtual Link
4. Bandwidth allocation per Virtual Link



AFDX Implementation with $P4_{14}$

1. Frame Format

- Based on Ethernet frame
- Virtual Link encoded in destination MAC address

```
header_type afdx_t {
  fields {
    dstConst : 32;
    dstVlinkID : 16;
    srcAddr : 48;
    etherType : 16;
  }
}
header afdx_t afdx;
```

AFDX Implementation with $P4_{14}$

2. Integrity Checks

- Header format
- Frame size per Virtual Link

Ingress Function

```
control ingress {
  integrity_check();
  if(afdx.dstConst == DST_CONST) {
    apply(tbl_forward_virtual_link);
    traffic_policing();
  } else {
    apply(do_drop);
  }
}
```

AFDX Implementation with $P4_{14}$

3. Static Forwarding per Virtual Link

- Vendor-specific multicast

```
action forward(egress_ports) {
{
  modify_field(standard_metadata.egress_spec,
    EGRESS_SPEC_MULTICAST);
  modify_field(intrinsic_metadata.egress_port_bitmap,
    egress_ports);
}

table tbl_forward_virtual_link {
  reads {
    standard_metadata.ingress_port : exact;
    afdx.dstVlinkID                : exact;
  }
  actions {
    drop;
    forward;
  }
  size : MAX_VIRTUAL_LINKS;
}
```

AFDX Implementation with $P4_{14}$

4. Bandwidth Allocation per Virtual Link

AFDX terminology: Filtering based on time between frames

- Minimum time between first bit of consecutive frames
- In AFDX called Bandwidth Allocation Gap

Implementation 1: Ingress timestamp & egress scheduler

- Requires egress time for egress scheduler

→ Additional support by P4 switch required

Implementation 2: Bandwidth limitation of Virtual Link

- Possible with P4 meter

```
meter vlink_bandwidth {
  type   : bytes;
  direct : tbl_forward_virtual_link;
  result : scheduling_metadata.color_bytes;
}

control traffic_policing {
  if(scheduling_metadata.color_bytes != COLOR_GREEN) {
    apply(do_drop);
  }
}
```

AFDX Implementation with $P4_{14}$

Performance

Prototypes

- Software: P4@ELTE
- FPGA: Xilinx Zynq
- Network Flow Processor (NPU): Netronome Agilio

Switch	Latency (1500 B)
Rockwell Collins AFDX switch [1]	5 µs
HP E3800 without OpenFlow [1]	7,2 µs
HP E3800 with OpenFlow [1]	7,7 µs
HP E3800 with software switching [1]	613 µs (avg.)
P4 switch with P4@ELTE [2]	24 µs
P4 switch with NPU with CPU [2]	24 µs
P4 switch with NPU without CPU [2]	5,8 µs
P4 switch with FPGA [2]	1,2 µs

P4 switches competitive with existing hardware for Avionic Networks

[1] P. Heise, F. Geyer, and R. Obermaisser, “Deterministic OpenFlow: Performance Evaluation of SDN Hardware for Avionic Networks,” in Proceedings of the 11th International Conference on Network and Service Management (CNSM), Nov. 2015

[2] F. Geyer and M. Winkel, "Towards Embedded Packet Processing Devices for Rapid Prototyping of Avionic Applications." 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018). 2018.

P4 Enhancements for Avionics Use Cases

Scheduling

- Limited support (Strict Priority Queuing)
- Additional support desirable, for example: Weighted Fair Queuing [1] or Deficit Round Robin [2]

→ Required for QoS architectures

→ Programmable packet scheduling in parallel to P4 [3]

Time-based and Time-triggered Protocols

- No primitives to access clocking information
- Required for packet timestamping, egress scheduling based on timing information
- Cf. IEEE TSN: Time Sensitive Networking

→ Interface for retrieving time information

→ Vendor-specific and vendor-independent metadata

[1] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queueing Algorithm,” ACM SIGCOMM Comput. Commun. Rev., Aug. 1989
[2] M. Shreedhar and G. Varghese, “Efficient Fair Queuing Using Deficit Round-Robin,” IEEE/ACM Trans. Netw., Jun. 1996
[3] A. Sivaraman, S. Subramanian, M. Alizadeh, S. Chole, S.T. Chuang, A. Agrawal, H. Balakrishnan, T. Edsal, S. Katti, N. McKeown, "Programmable packet scheduling at line rate." Proceedings of the 2016 ACM SIGCOMM Conference. ACM, 2016.

P4 Enhancements for Avionics Use Cases

Device Certification for Safety Requirements

Improvements from $P4_{14}$ to $P4_{16}$ for Safety of Avionics Hardware

- Defined behavior (casting, exceptions, initial values)
- Portable Switch Architecture
- General: no loops
- Formal analysis and functional validation improved [1][2]

→ Verification of performance parameters

Goal: Certification of P4 hardware, tools and programs

[1] A. Nötzli, J. Kahn, A. Fingerhut, C. Barrett, P. Athanas, "P4pktgen: Automated test case generation for p4 programs." Proceedings of the Symposium on SDN Research. ACM, 2018.
[2] C. Cascaval, N. Foster, W. Hallahan, J. Lee, J. Liu, N. McKeown, C. Schlesinger, M. Sharif, R. Soulé, H. Wang, "p4v: Practical Verification for Programmable Data Planes", Proceedings of the 2018 ACM SIGCOMM Conference. ACM, 2018.

Conclusion

- P4 suited for prototyping in other domains
- Implementation of (only) required features
- Reduced development time and cost
- Requires expertise in P4 and target switch
- Requires better support for scheduling and timing information
- Goal: Certification of P4 hardware, tools and programs

Relevant full paper:
F. Geyer, M. Winkel, "Towards Embedded Packet Processing Devices for Rapid Prototyping of Avionic Applications"
In: 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), February, 2018
`https://hal.archives-ouvertes.fr/hal-01711011/document`