Low-latency in-network applications through programmable data planes

PhD candidate: Fabricio Rodriguez
Advisor: Prof. Christian Esteve Rothenberg

07/10/2024

feec
INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP
UNICAMP

Agenda

01. Introduction

- Motivation
- Research Questions
- Methodology

02. Industrial Robot Control

- In-network control

03. P4 Programmable Patch Panel (P7)

- Link characteristics
- Multiple pipelines

04. In-network Collision Avoidance

- Collision Avoidance Algorithms
- P4 approach

05. P7 + UAV

- P4 Collision Avoidance using P7

06. Final Remarks

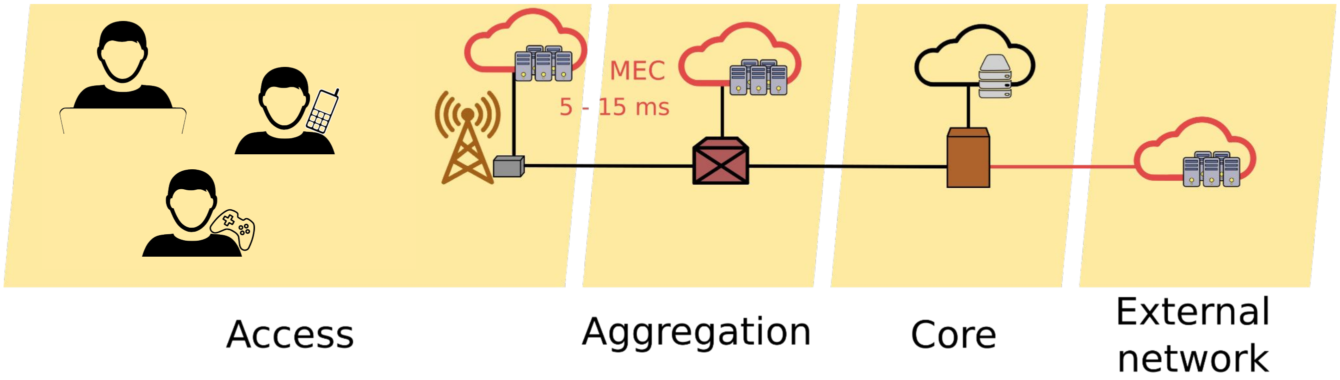
- Contributions & publications
- Conclusion & Future Work

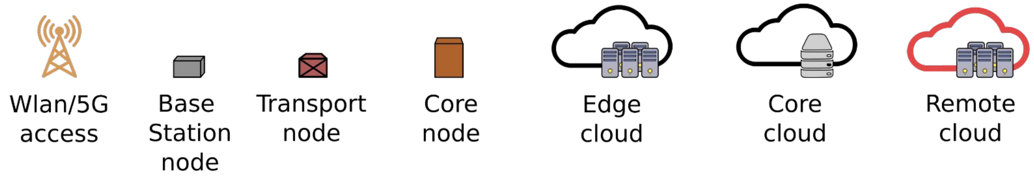
01.

Introduction

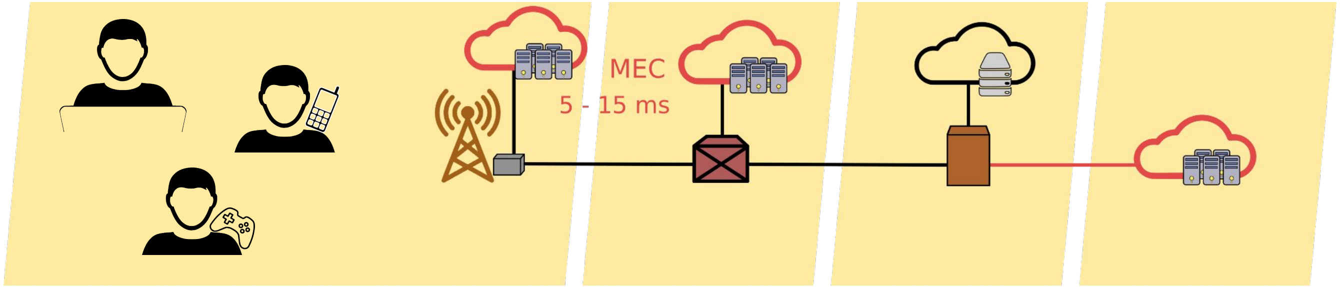
- Motivation
- Research Questions
- Methodology

Traditional network environment





IoT era



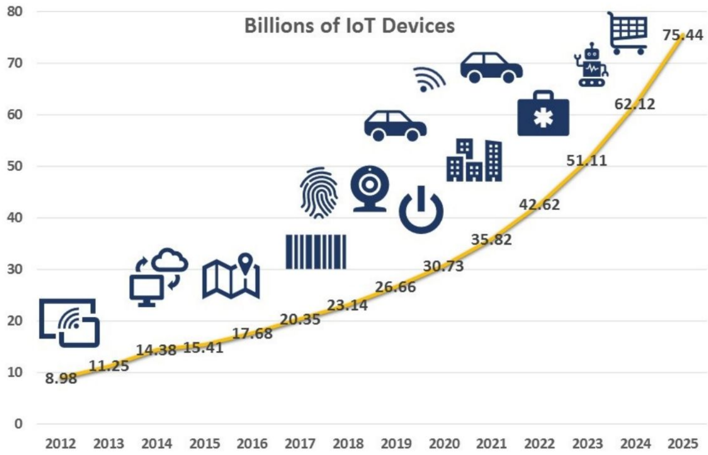
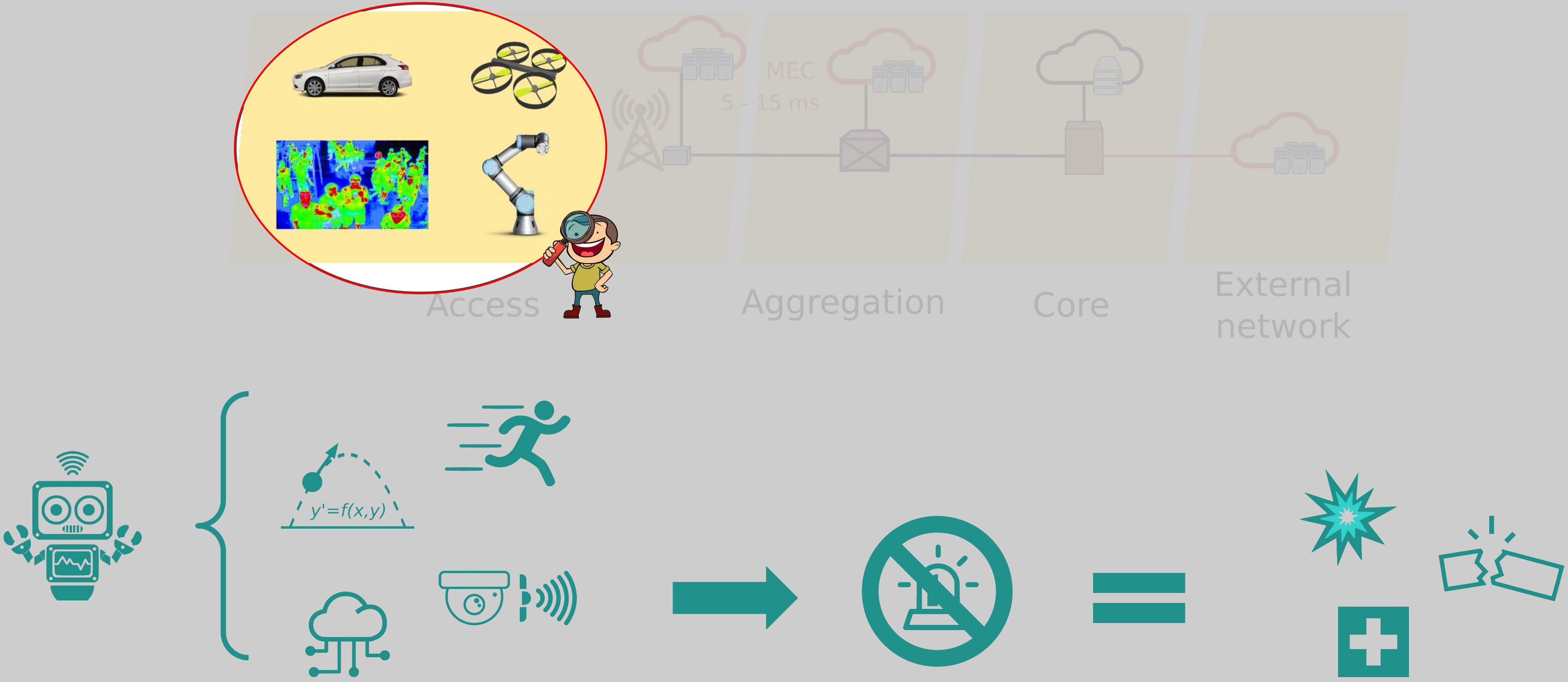
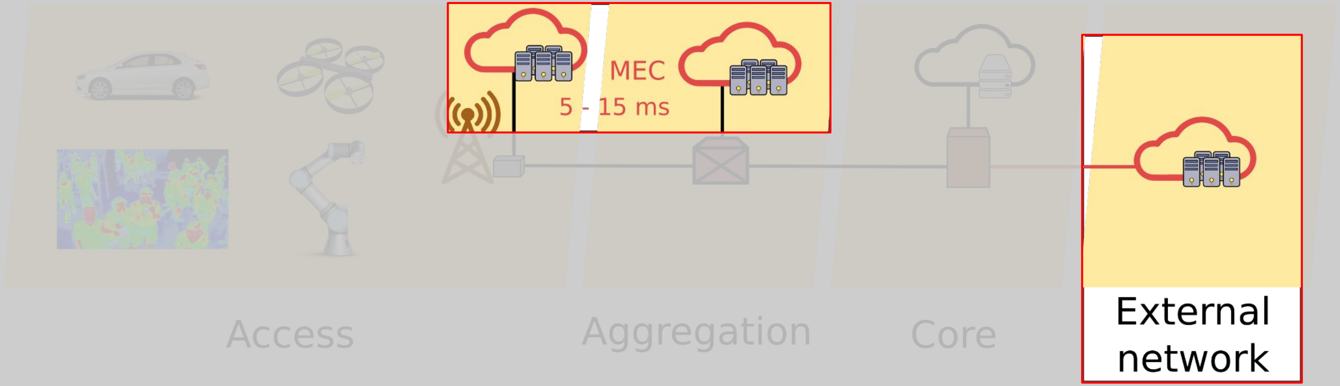


Figure source: Toor, Waqas & Alvi, Maira & Agiwal, Mamta. (2020). Combined Access Barring Scheme for IoT Devices Using Bayesian Estimation. Electronics. 9. 2191. 10.3390/electronics9122191.

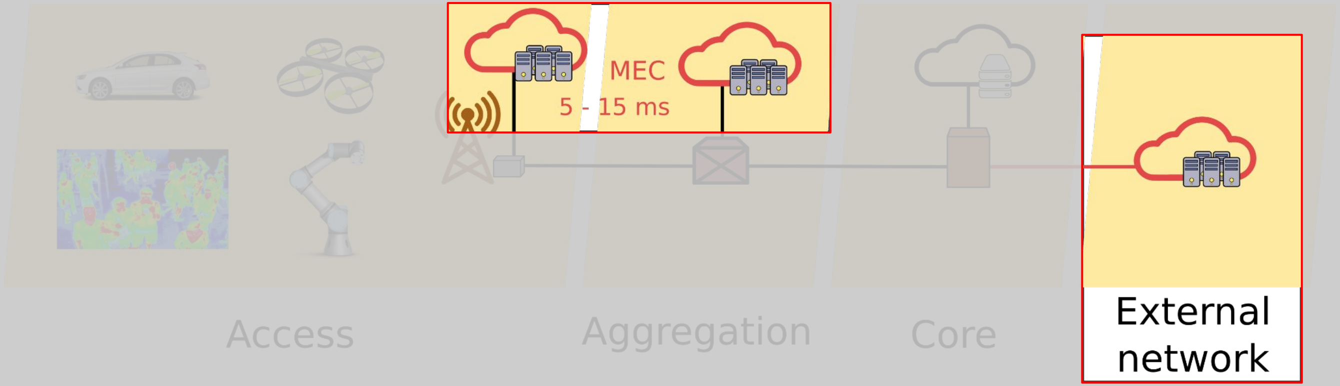
New use cases

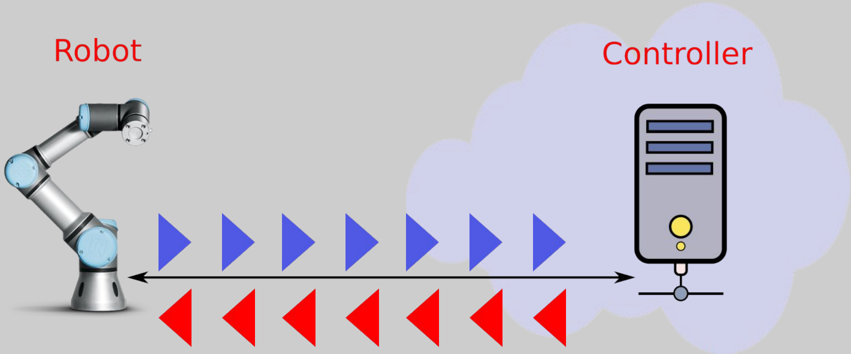


Cloud controllers

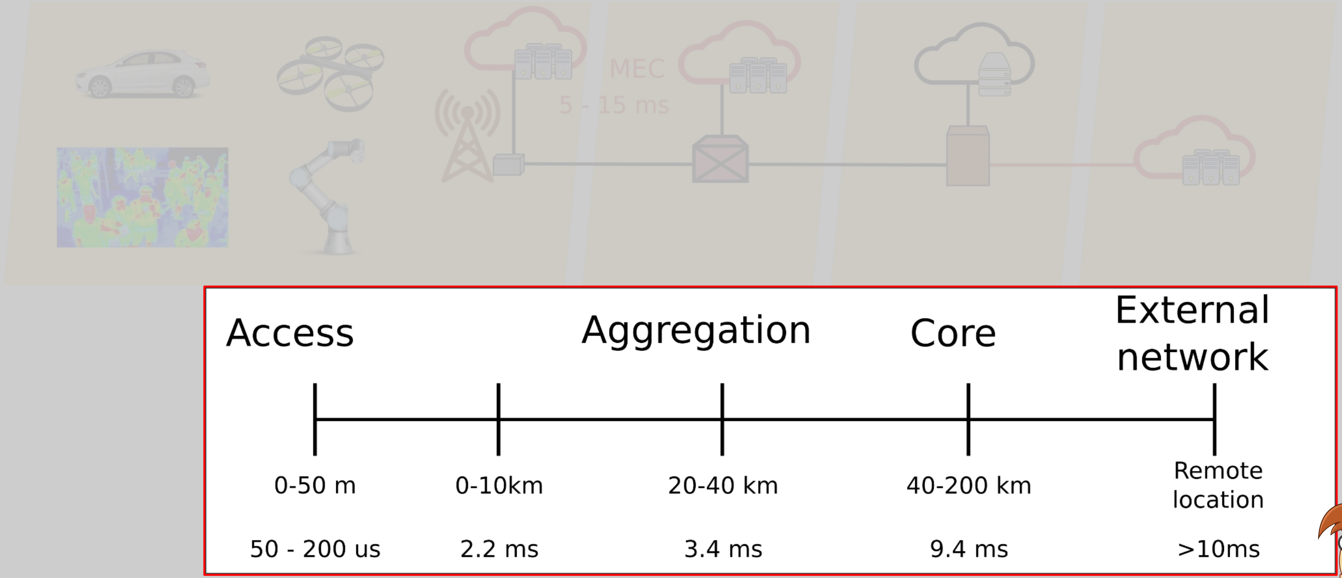


Cloud controllers



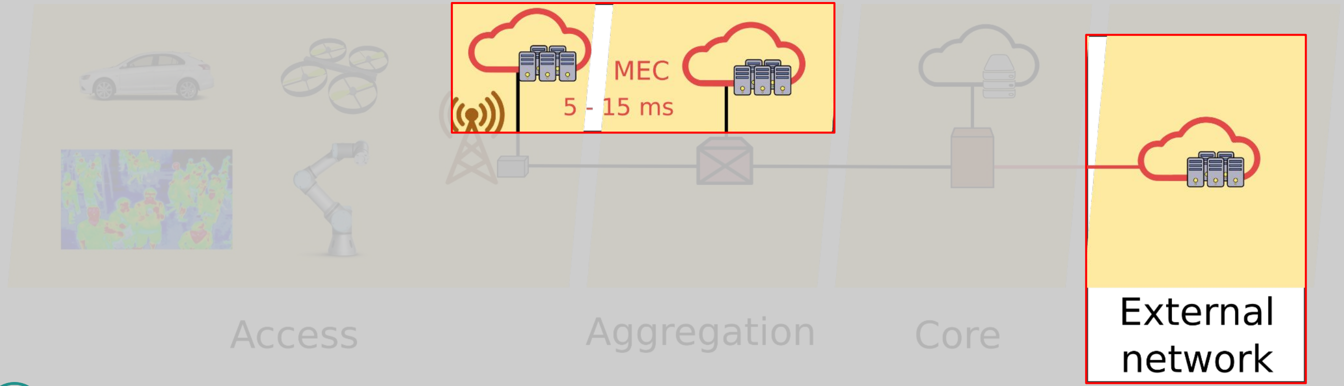


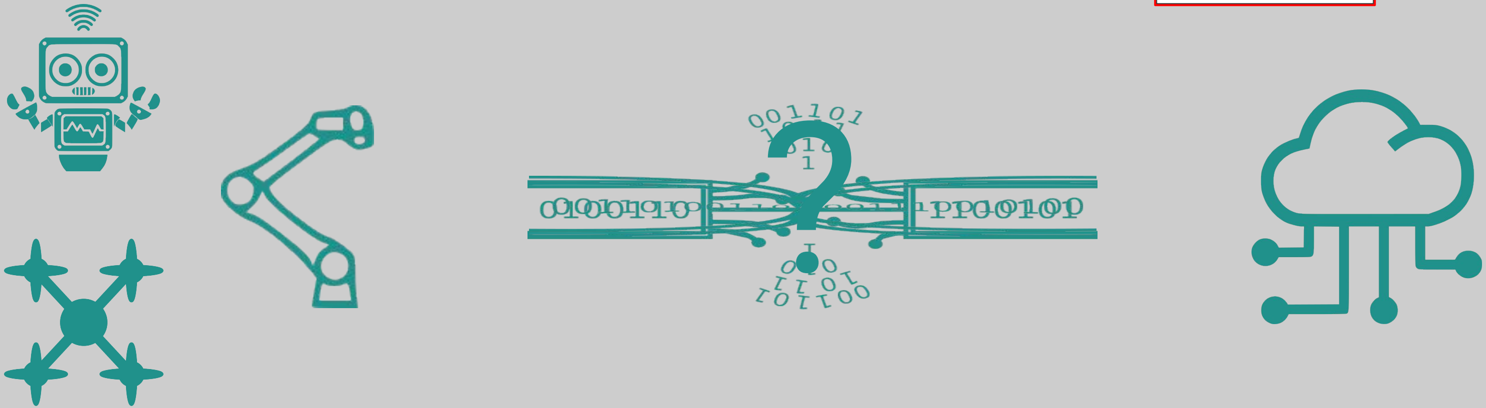
Horizontal delays



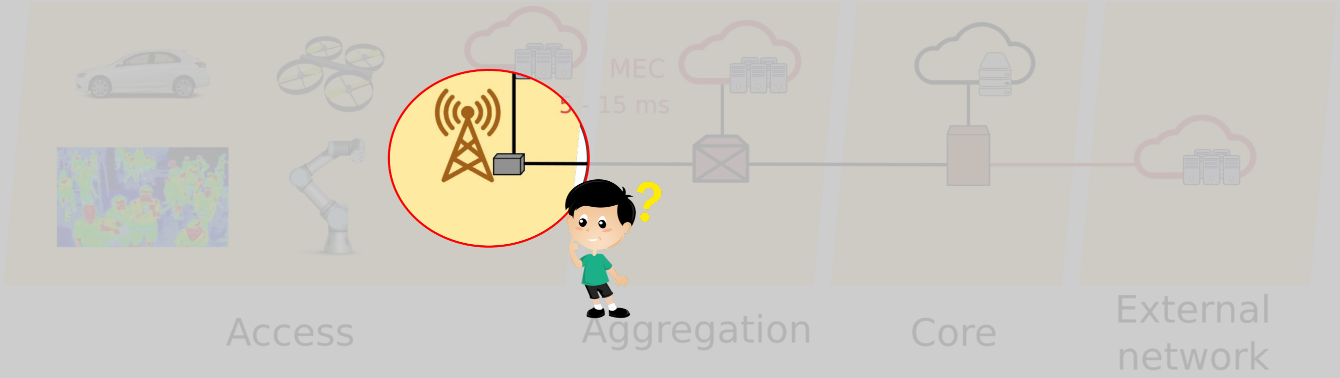
Source: Brown, Gabriel. "New Transport Network Architectures for 5G RAN". Heavy Reading.
Source: Sutton, Andy. "5G Network Architecture, Design and Optimisation". 5G - State of Play.
Source: Camps, Daniel. "Impact of 5G RAN Architecture in Transport Networks". ONDM

Cloud controllers

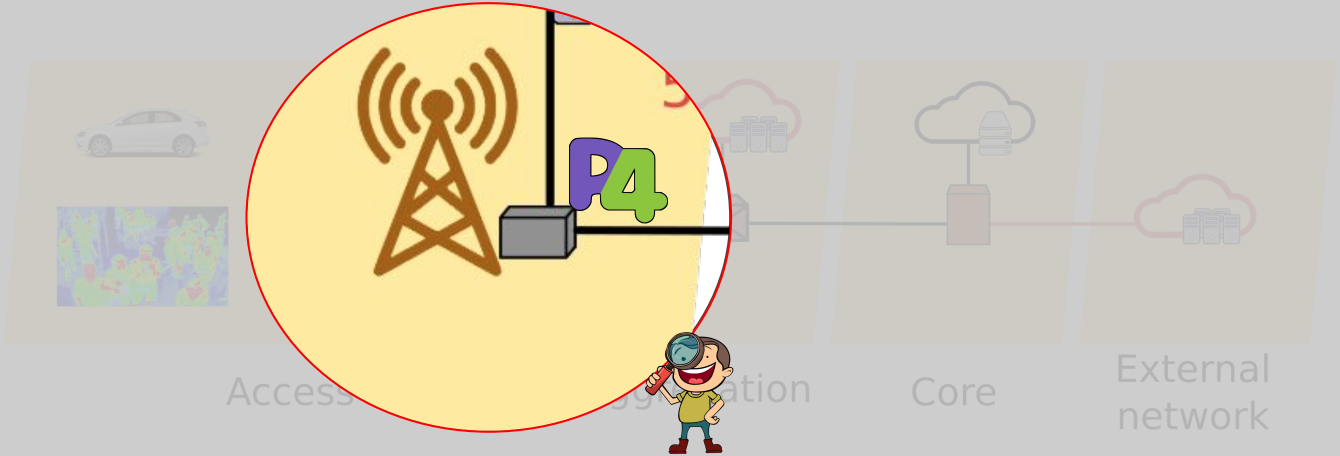




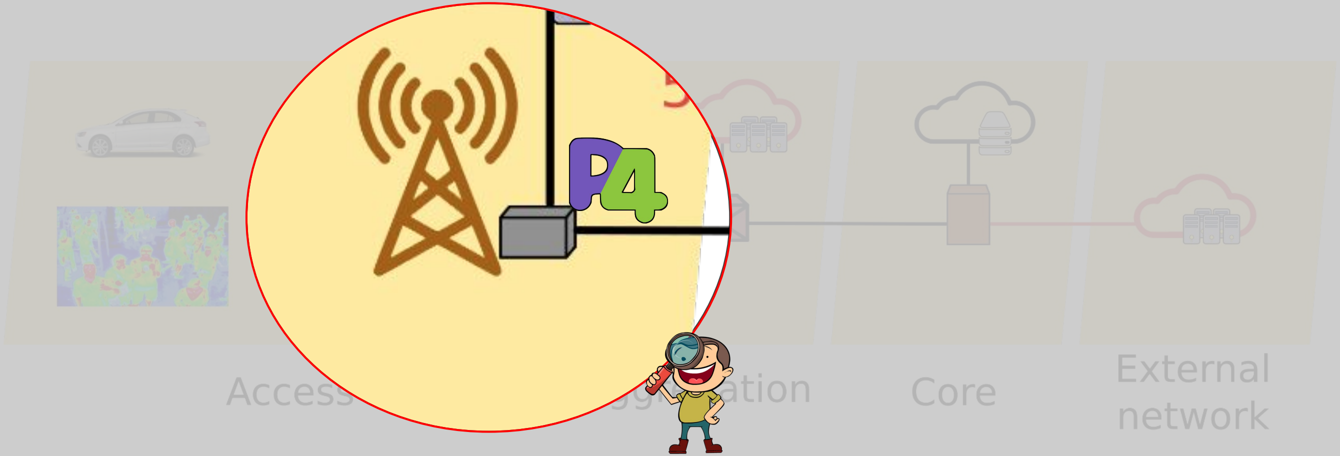
Edge solution?

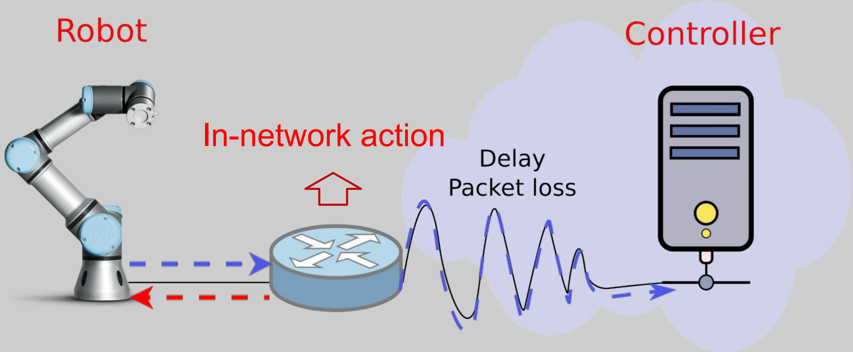


In-network solution?



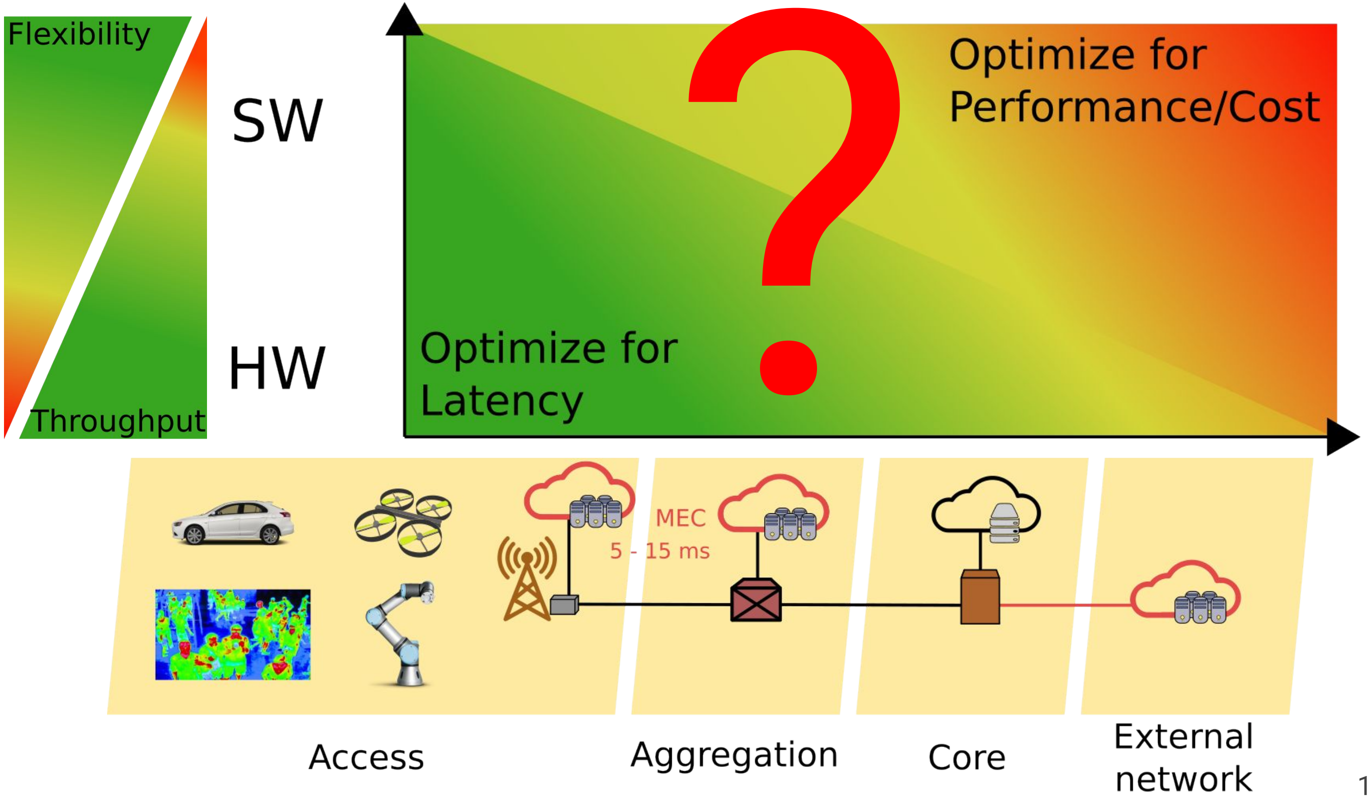
In-network solution?





Why In-network approach?

- Containers
- User Space
- Kernel Space
- Drivers, I/O SDKs
- General-purpose CPU
- HW-accelerated features
- FPGA
- GPU, TPU
- Programmable NIC, ASIC
- Domain specific architectures (P4+PISA)



Source: Christian Rothenberg. 2023. Fluid Network Control and Data Plane Research. Red Hat Research Days event.

Why In-network approach?

- ✓ Enable network programmability.
- ✓ Custom packet and pipeline distribution.
- ✓ Processing logic.
- ✓ New features.
- ✓ High throughput and low latency.

Why In-network approach?

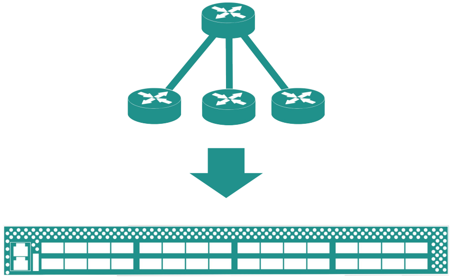
?

- Implement a custom algorithm ?
- Synchronize the communication ?
- Process and respond messages ?
- Integrate with other devices (e.g., robots and UAVs) ?

Why In-network approach?

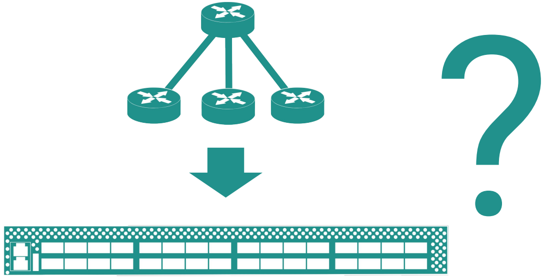
- Development of an in-network solution able to control a robot's arm movement allowing a rapid solution to avoid a possible collision.
- Evaluation of the in-network approach regarding its ability to respond on time and faster than a cloud or remote solution.

Why hardware-based emulation?



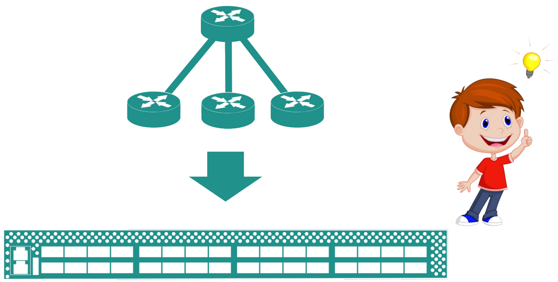
- ✓ Realistic experimental platform.
- ✓ High-fidelity performance, scalability, and flexibility.
- ✓ Data plane programmability
- ✓ P4-based application.

Why hardware-based emulation?



- Emulate network elements ?
- Emulate link metrics ?
- Support custom pipelines ?
- Integration with network controller ?

Why hardware-based emulation?

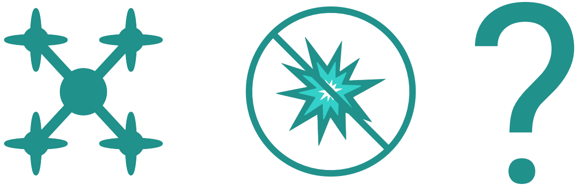


- Development of an environment suitable for network emulation allowing high-fidelity experiments with programmable data plane devices that replicate real networking scenarios.
- Provide source code and open tool to ensure reproducibility.

Why collision avoidance algorithm?

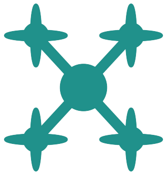
- ✓ Necessary to prevent collisions (e.g., damage or accidents).
- ✓ Improving the efficiency and safety of UAVs operations.
- ✓ Offload to the network.
- ✓ Integrate with network functions.

Why collision avoidance algorithm?



- Implement a P4-based algorithm ?
- Evaluate performance & scalability ?
- ULL response ?
- Centralized vs distributed ?

Why collision avoidance algorithm?



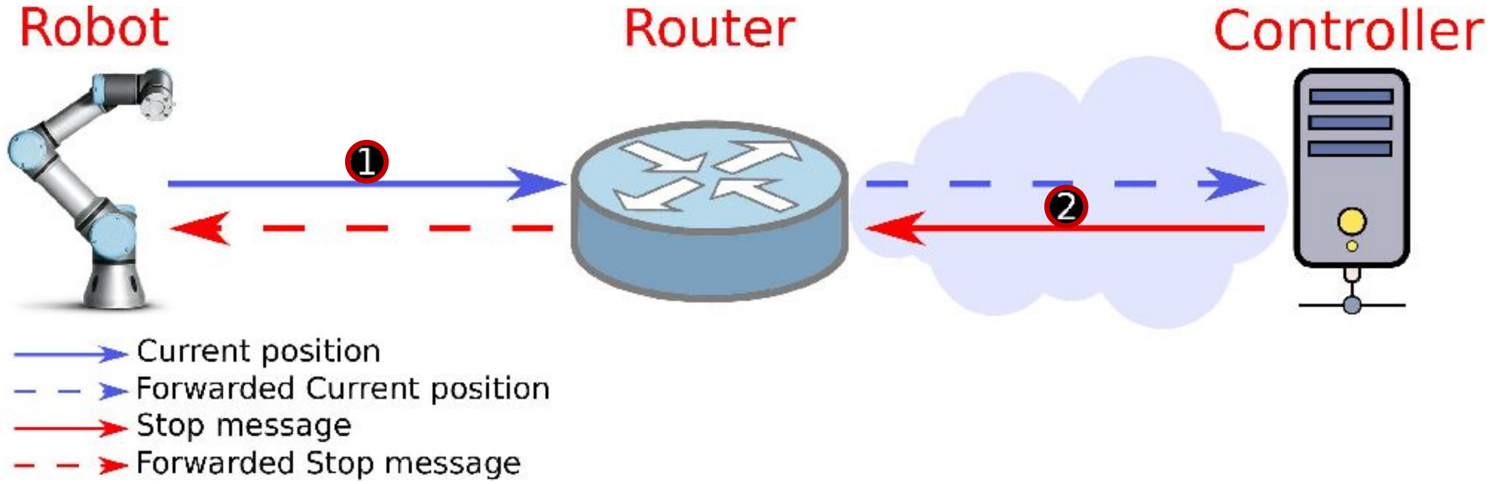
- Development of an in-network solution able to perform a collision avoidance task for a UAV.
- Conduction proof-of-concept experiments to validate our claims of a realistic and accurate collision avoidance algorithm in programmable data plane device.

02.

Industrial Robot Control

- In-network control

Traditional scenario without in-network control



In-network P4-based implementation

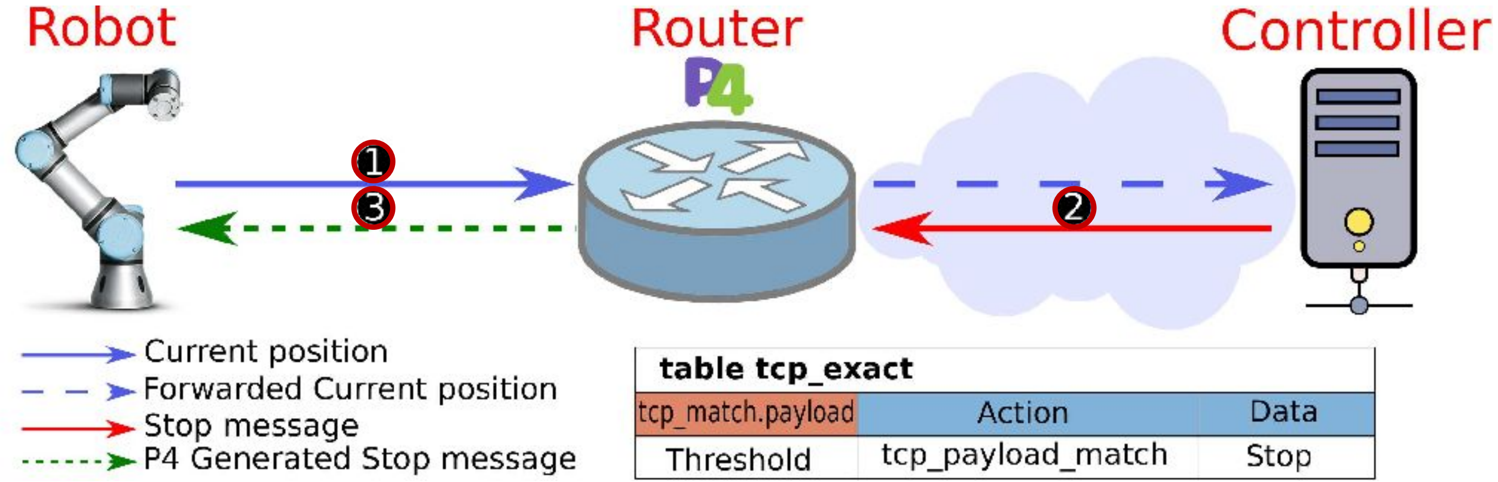
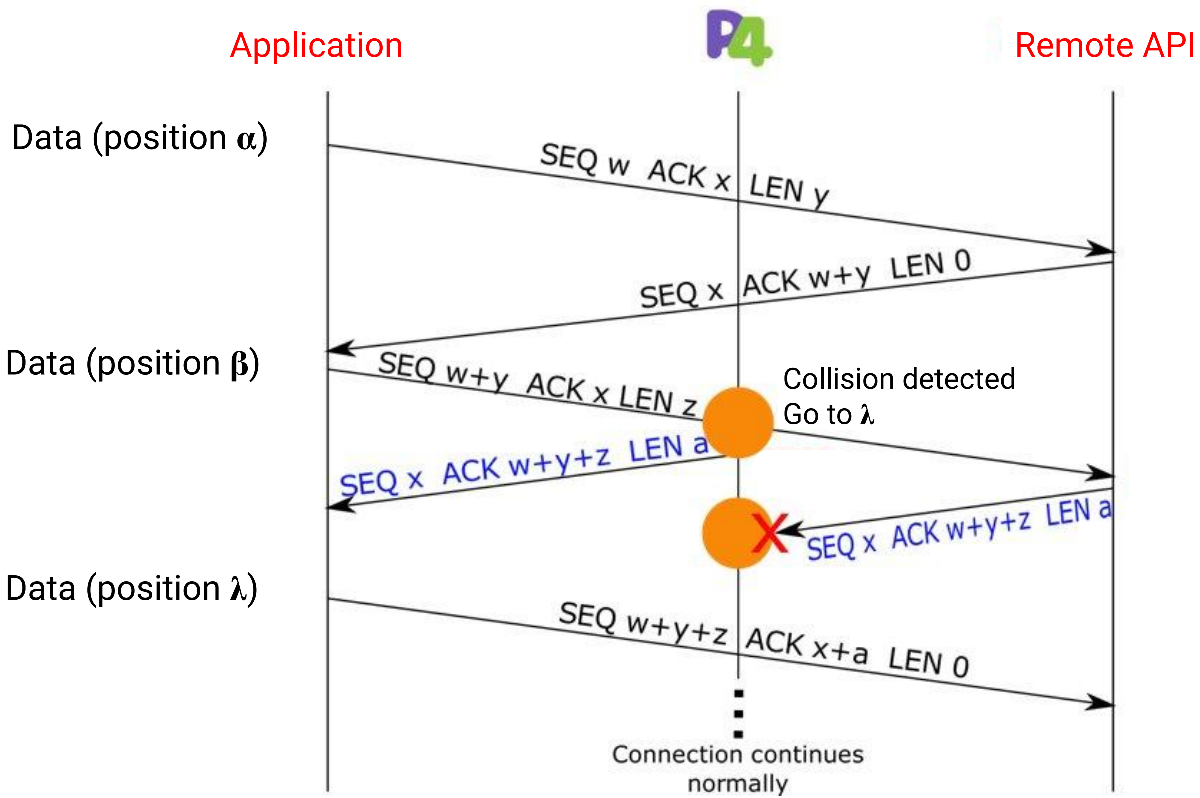


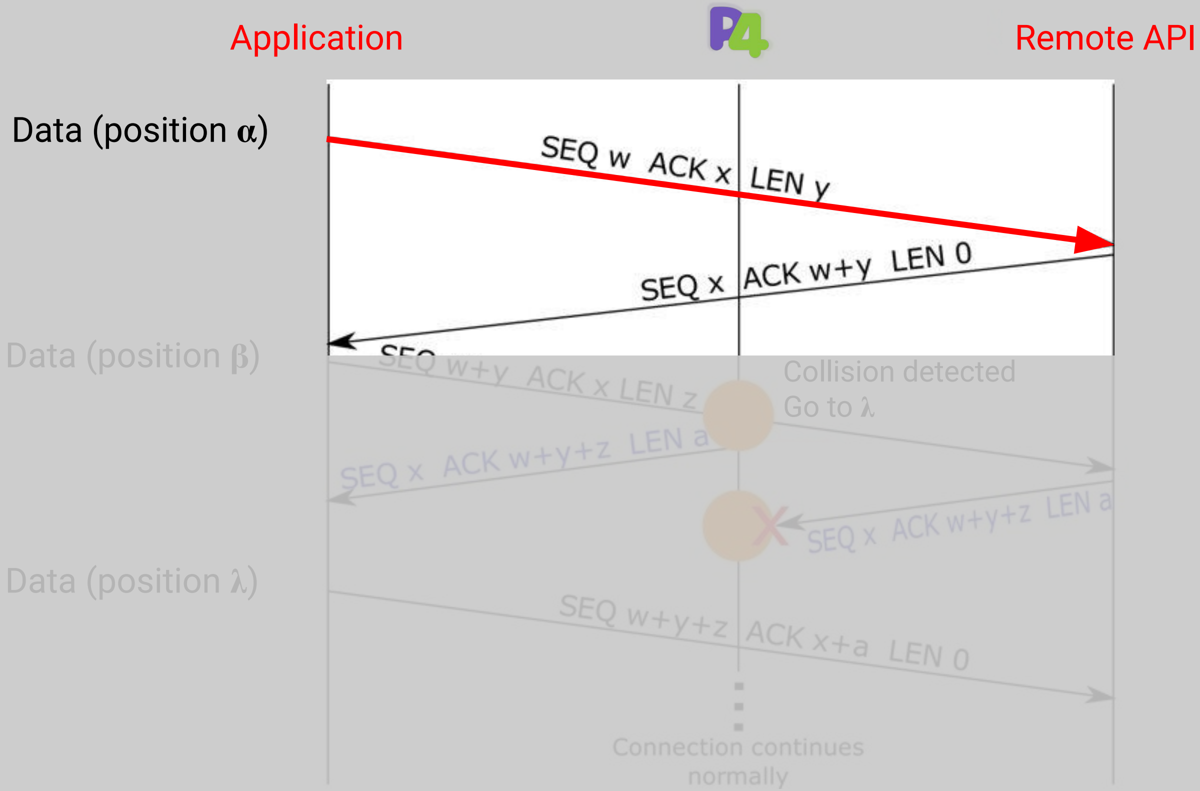
table tcp_exact

tcp_match.payload	Action	Data
Threshold	tcp_payload_match	Stop

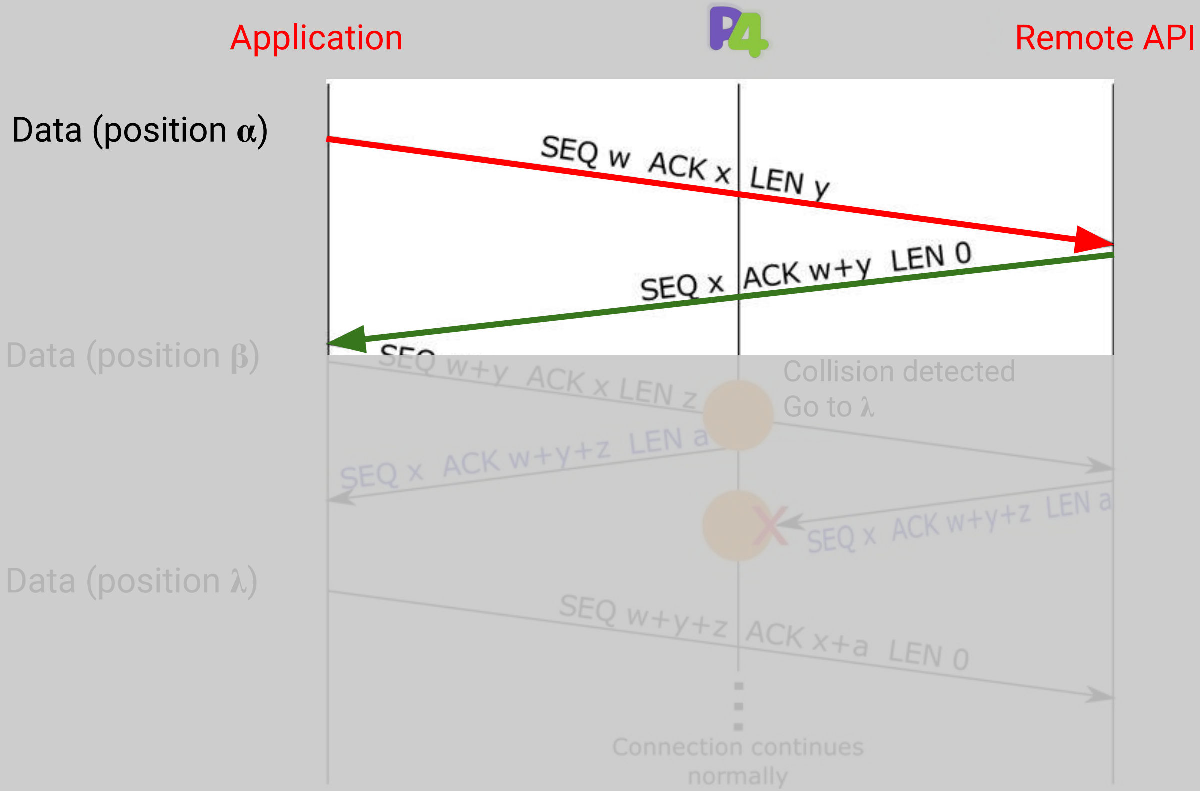
TCP session handler



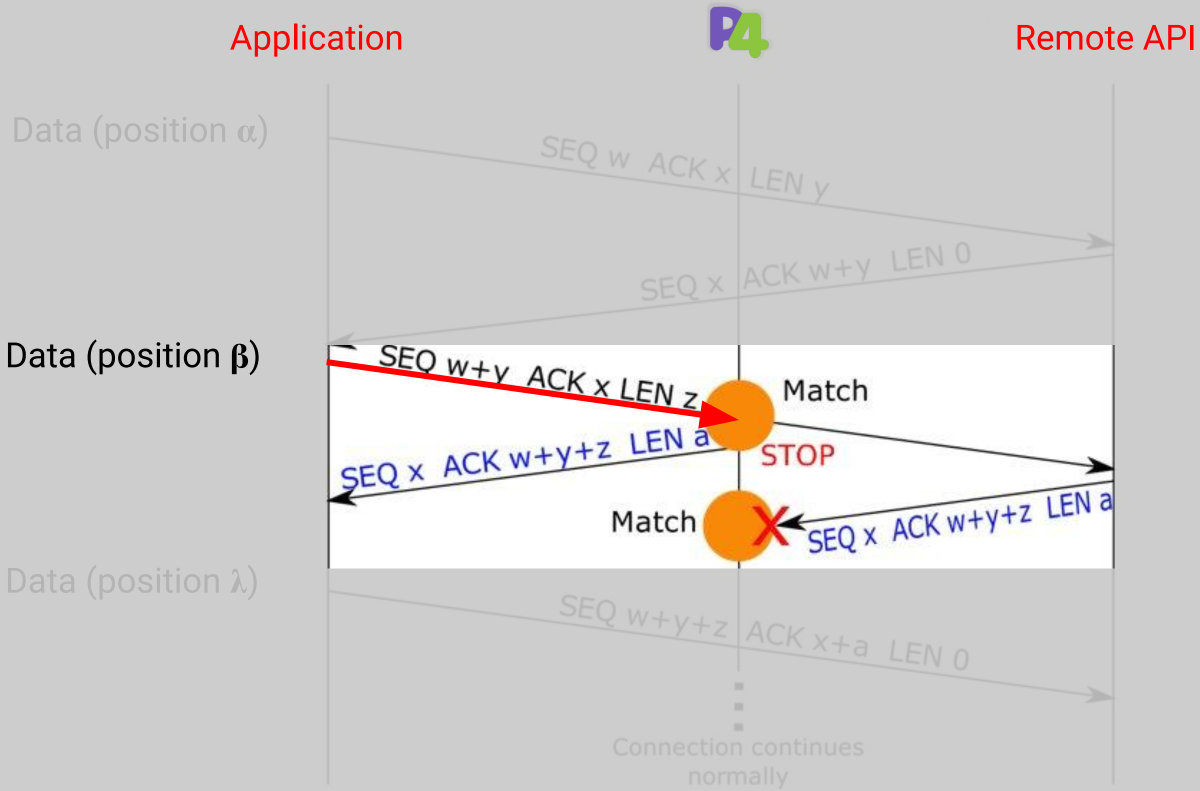
TCP session handler



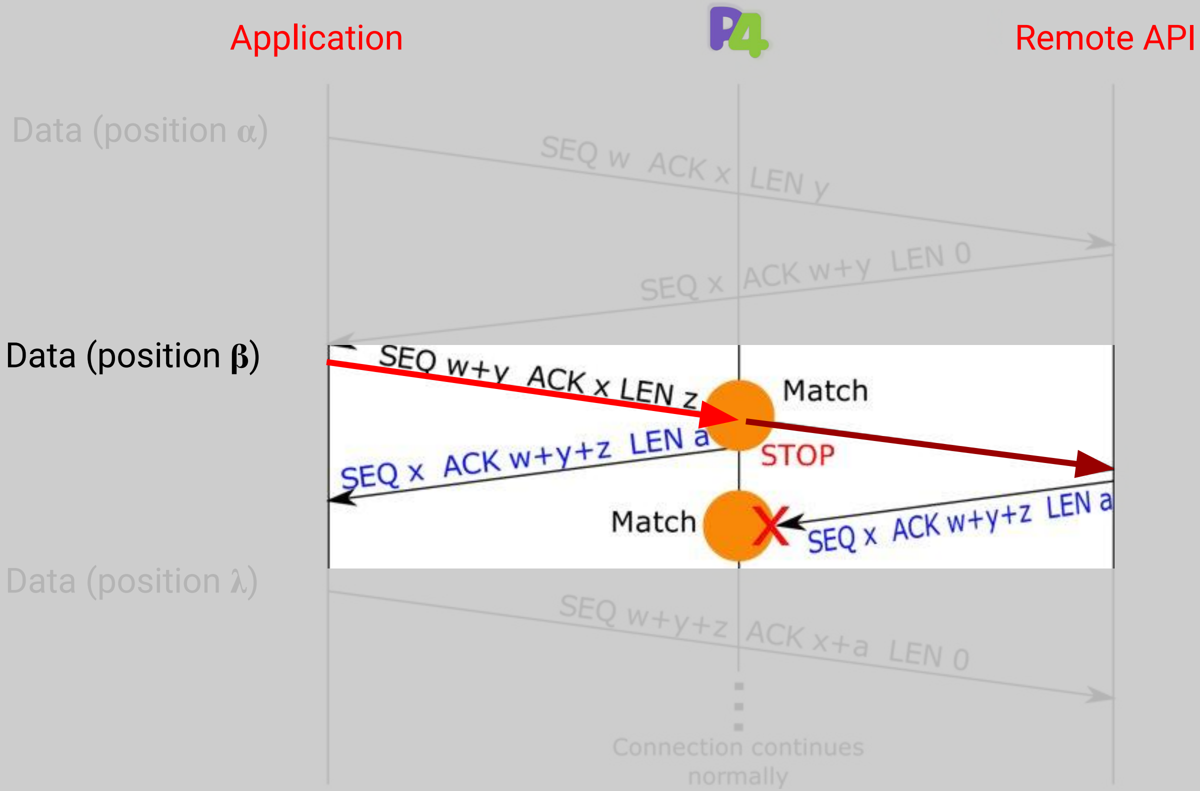
TCP session handler



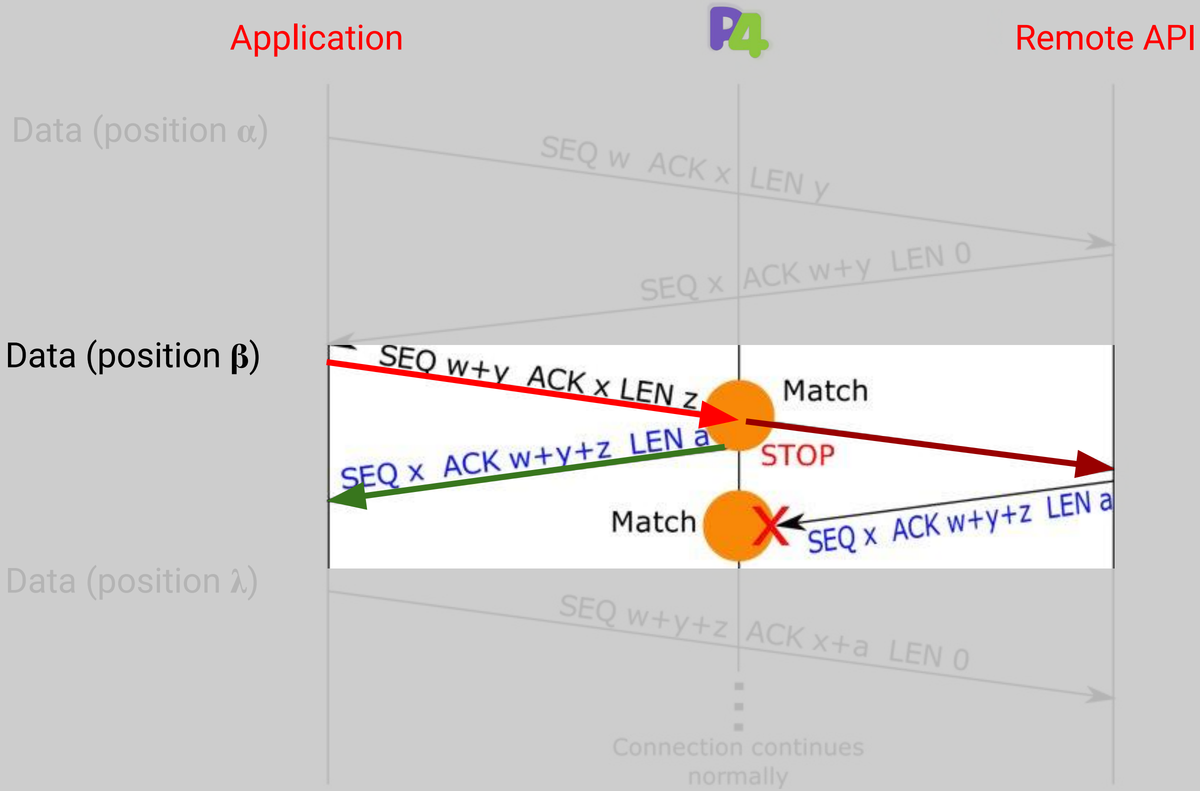
TCP session handler



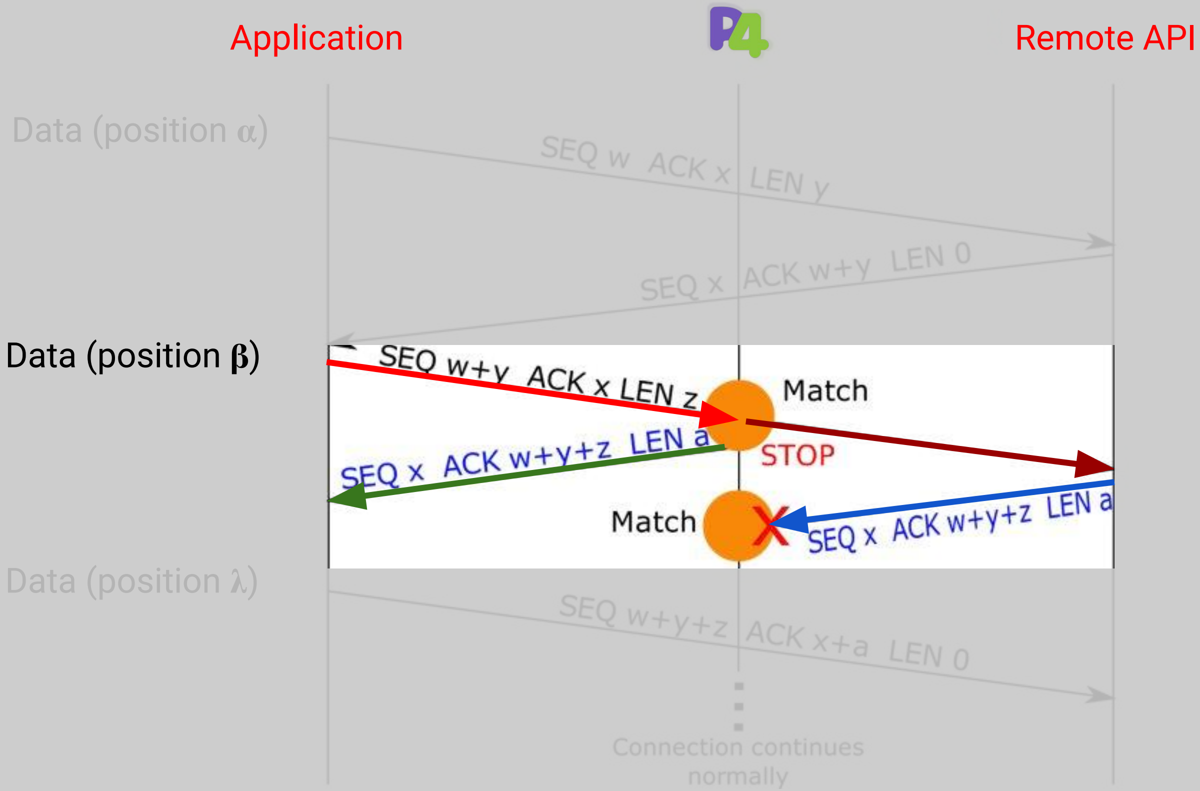
TCP session handler



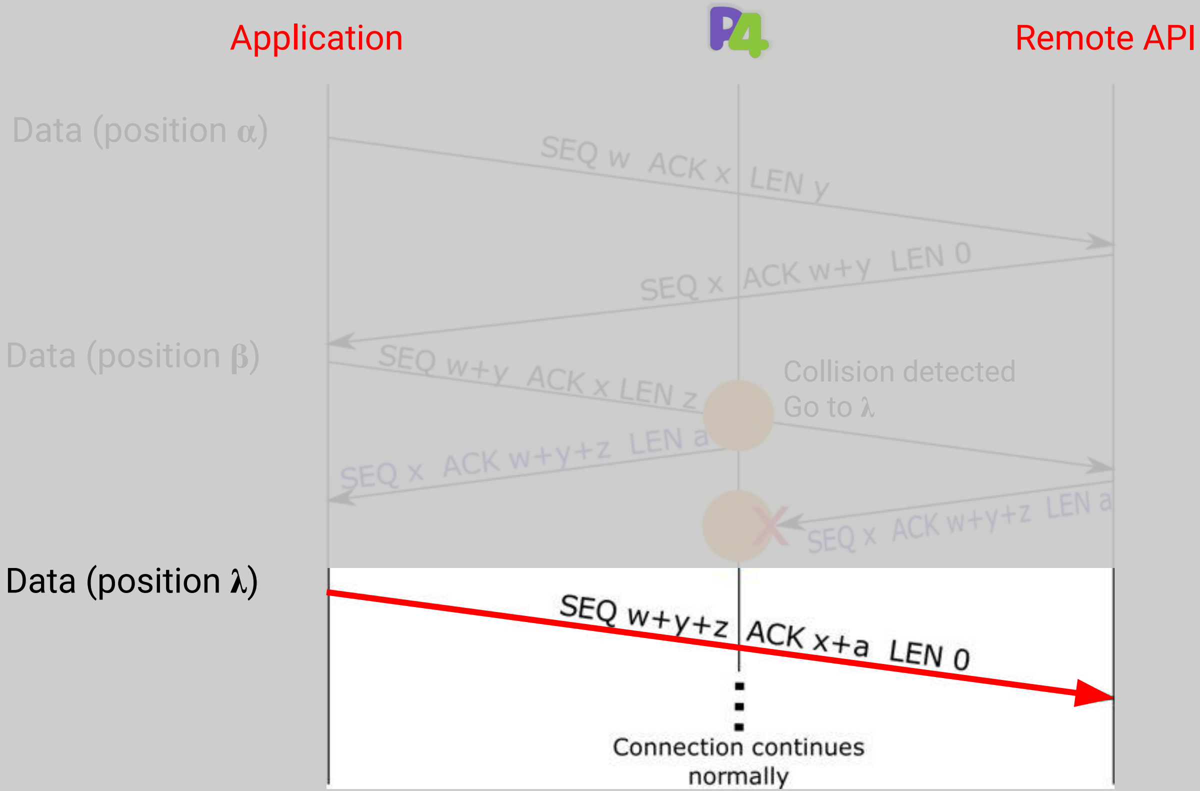
TCP session handler



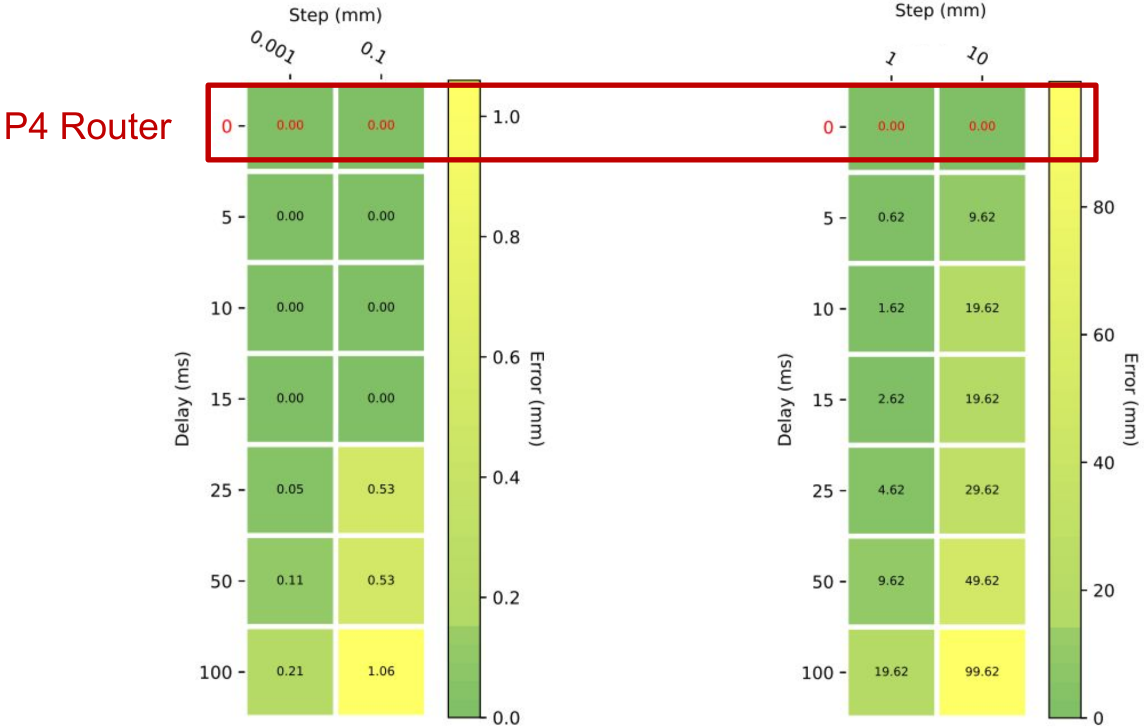
TCP session handler



TCP session handler

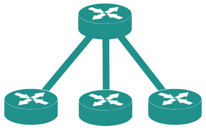


Stop position error



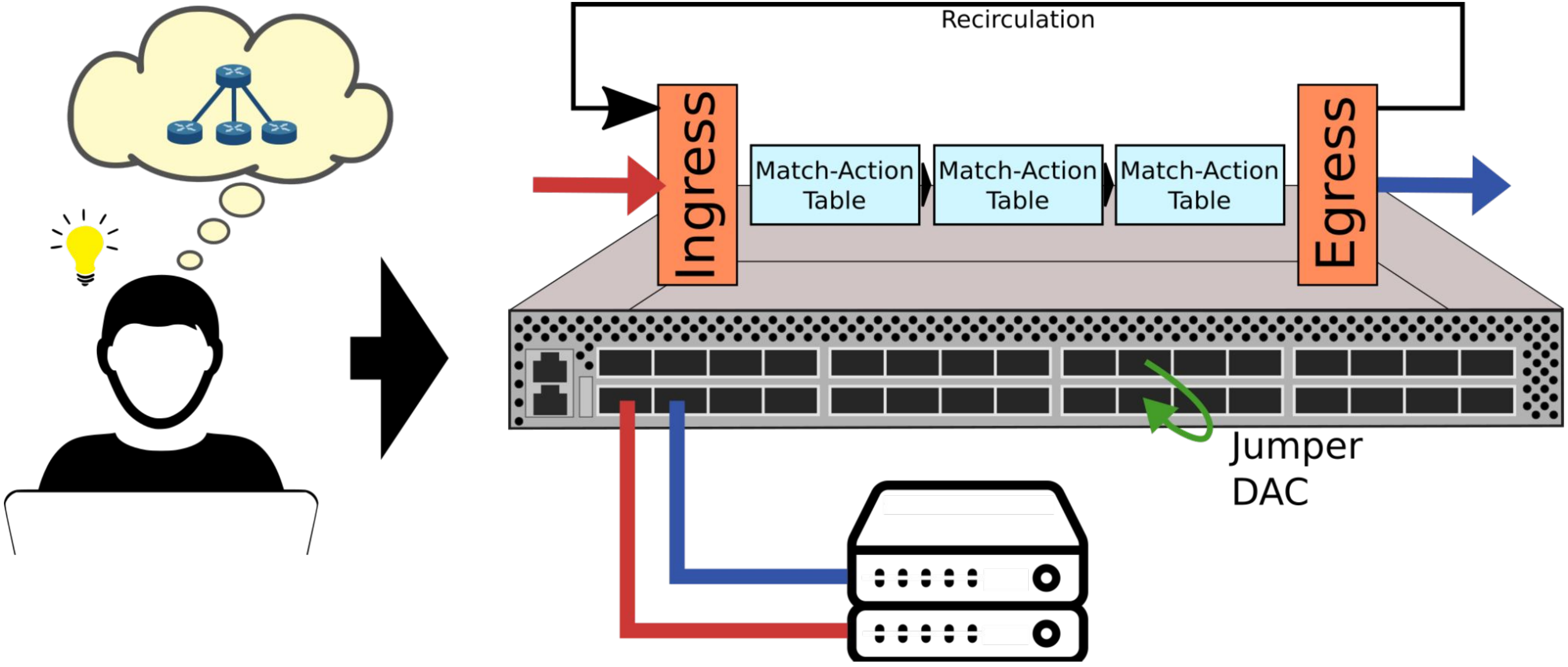
Steps of 0.001 - 0.1 mm

Steps of 1 - 10 mm

03. 

P4 Programmable Patch Panel (P7)

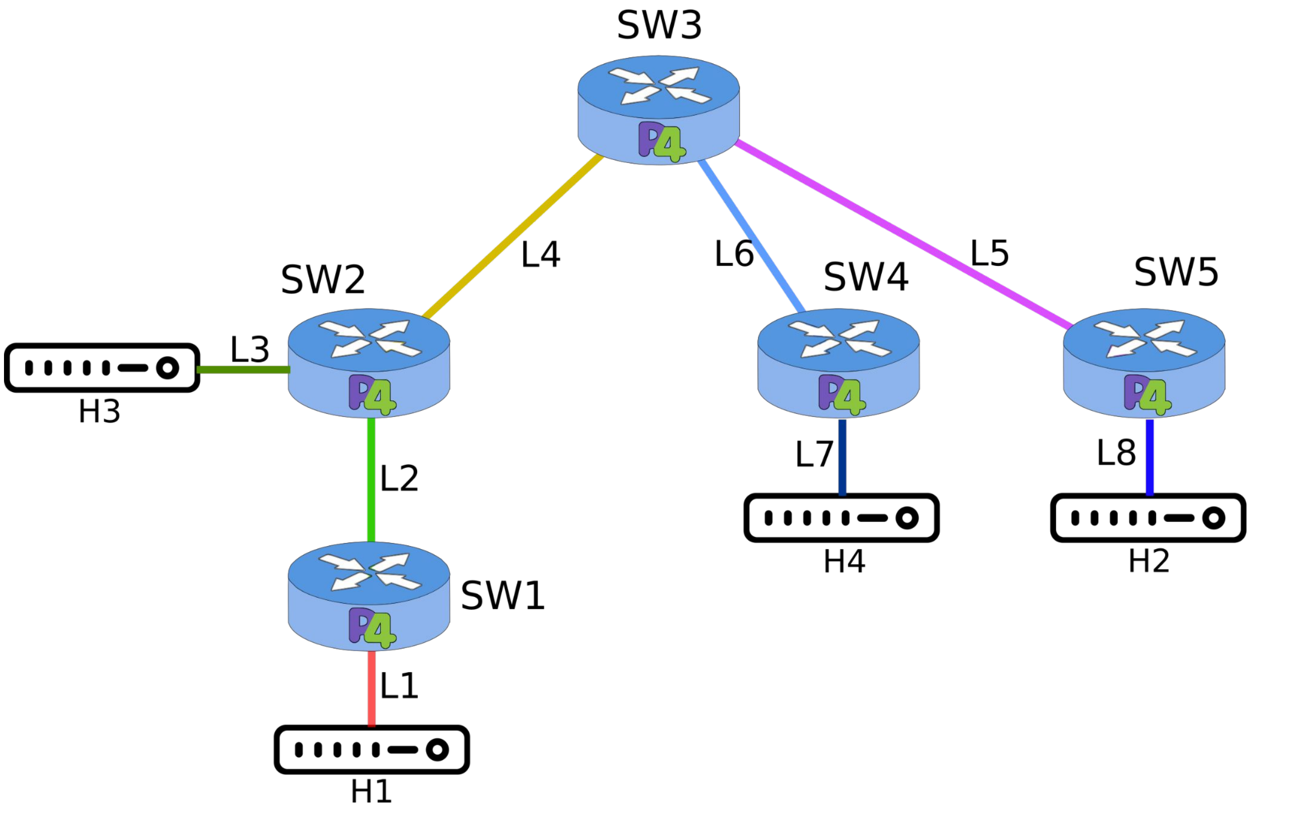
- Link characteristics
- Multiple pipelines


Recirculation
Ingress
Match-Action Table
Match-Action Table
Match-Action Table
Egress
Jumper DAC

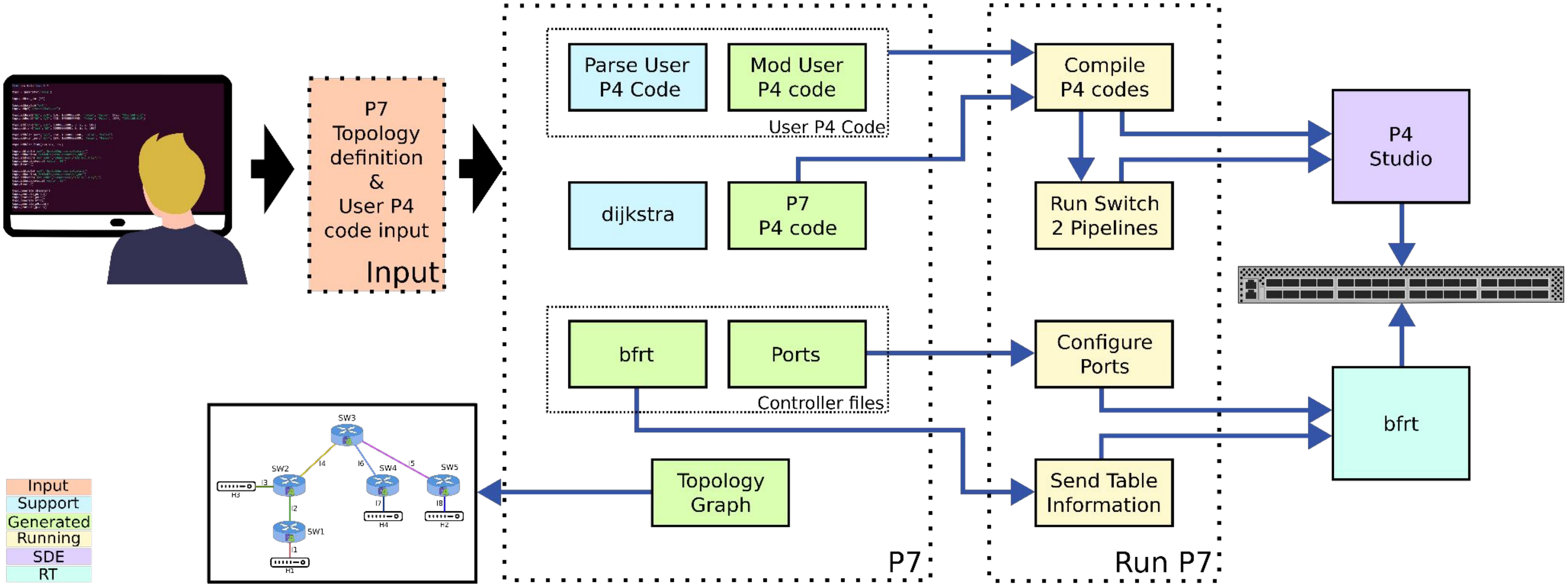
Link characteristics and approaches

Link Connectivity	Jumper cabling with/without internal Tag Intern Recirculation + internal Tag
Latency [ms]	Internal timer + recirculation TM + Pipelines recirculation
Jitter [ms]	Hash to determine recirculation times Lookup table with mathematical functions
Packet loss [%]	Random function to determine the probability to discard packets Realistic packet loss model
Re-ordering	TNA TM features Targeted probabilistic packet (ID) recirculation within a flow
Bandwidth	Rate limit TNA TM feature Port configuration and mapping

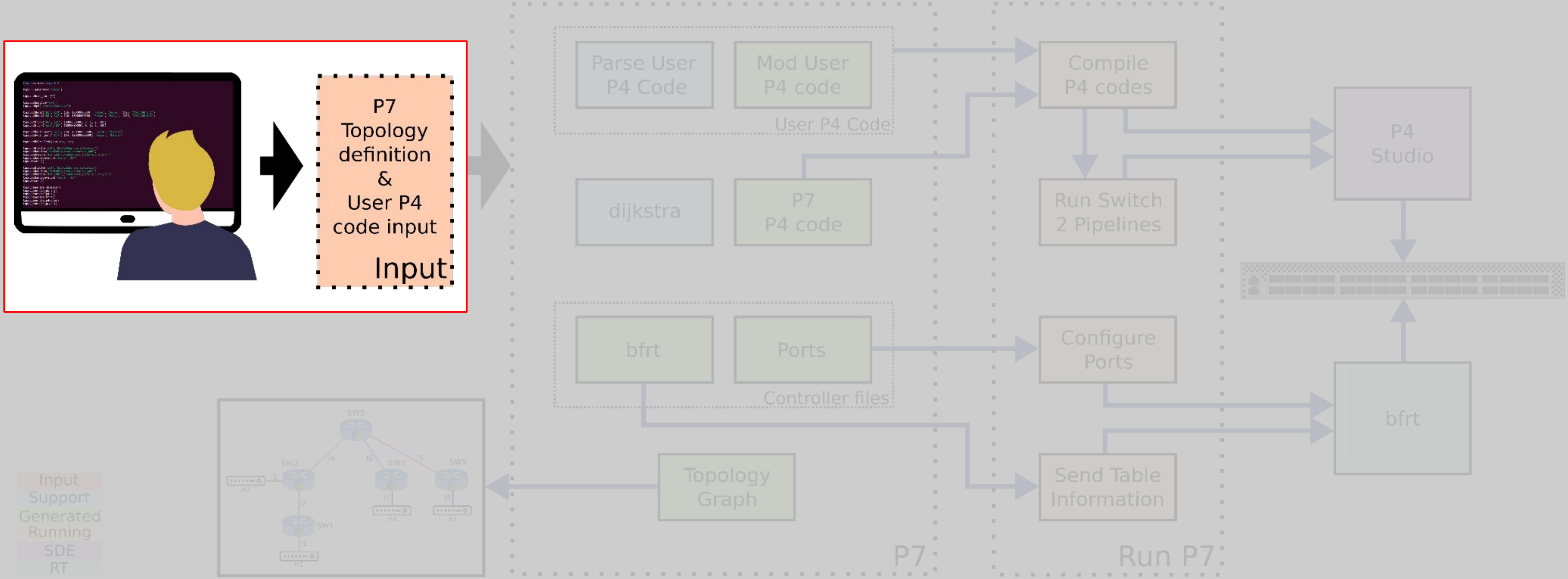
Network topology



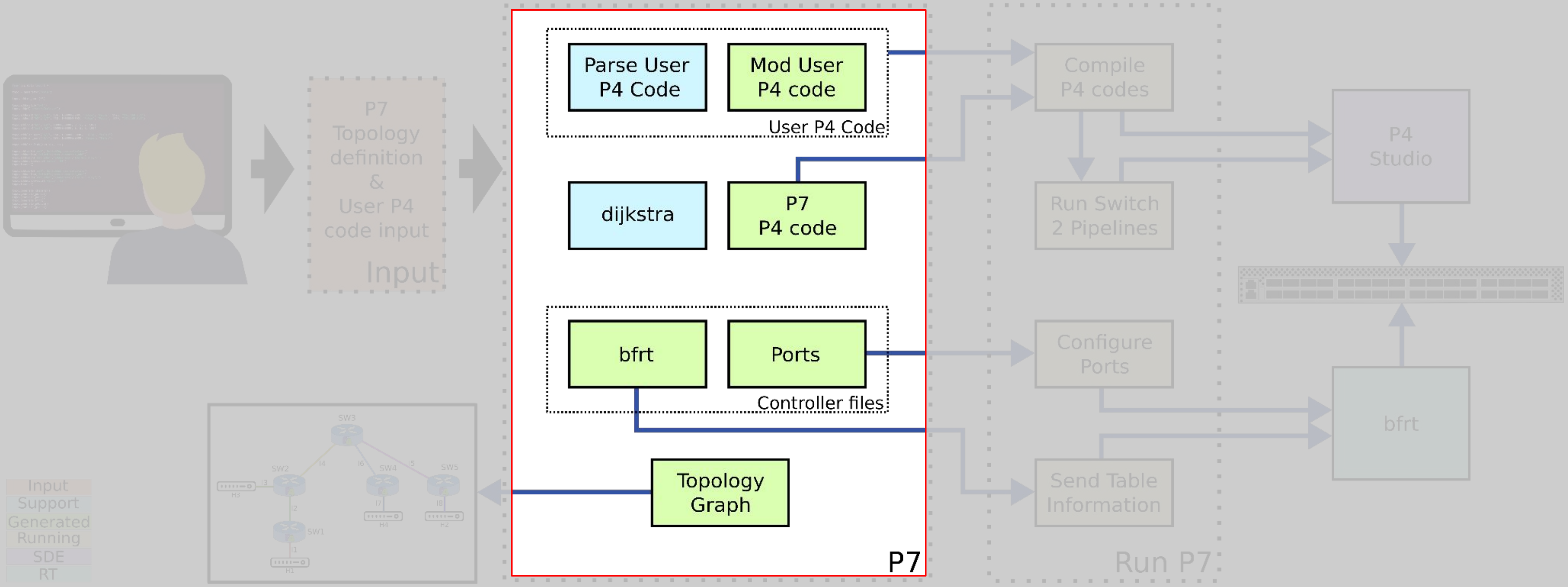
Architecture



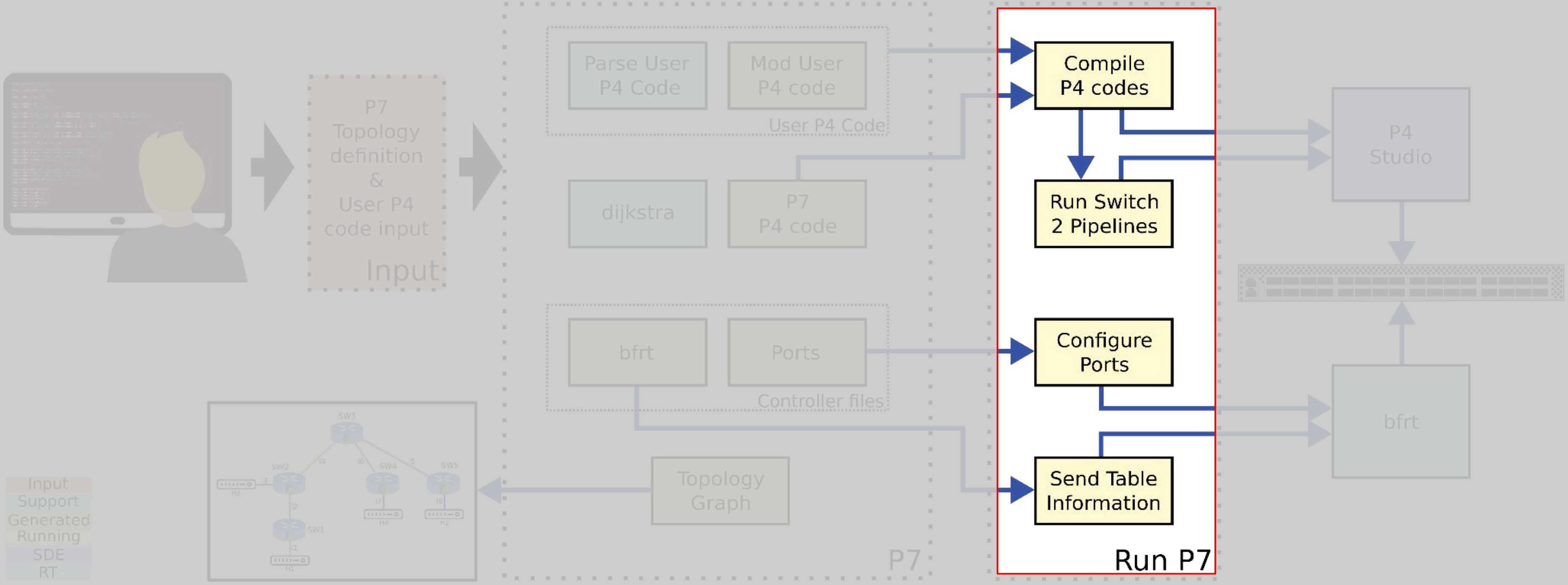
Architecture



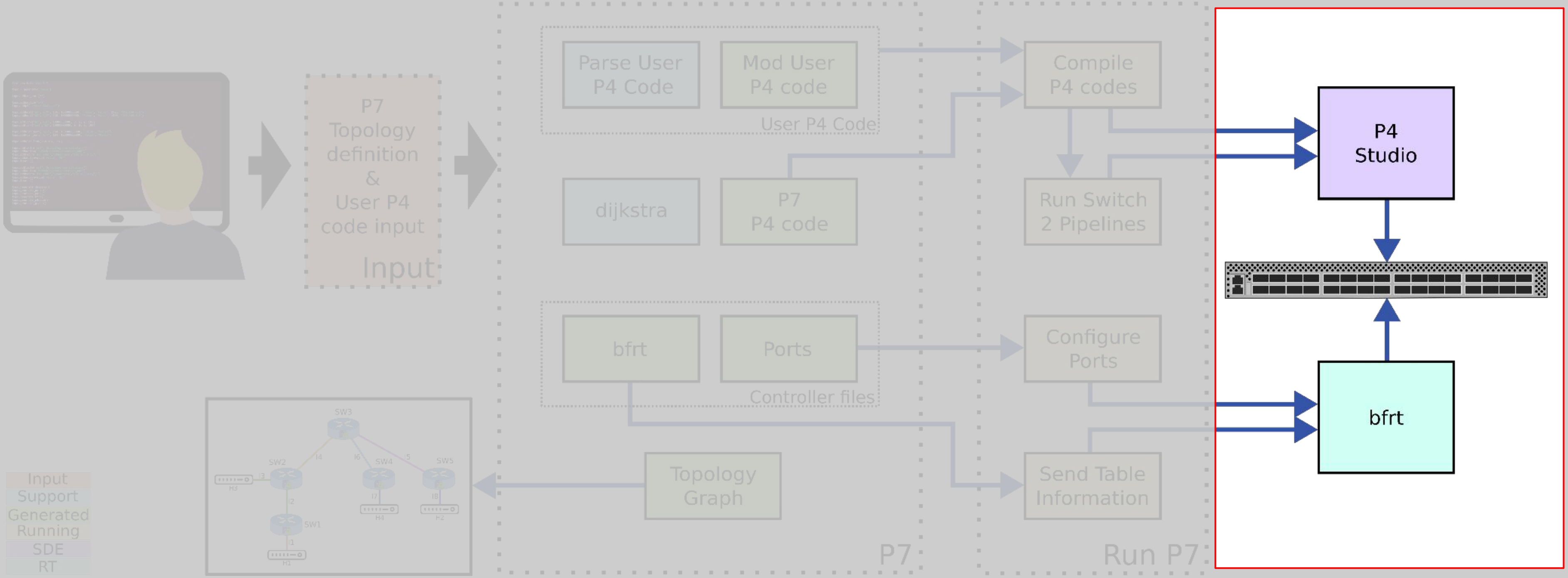
Architecture



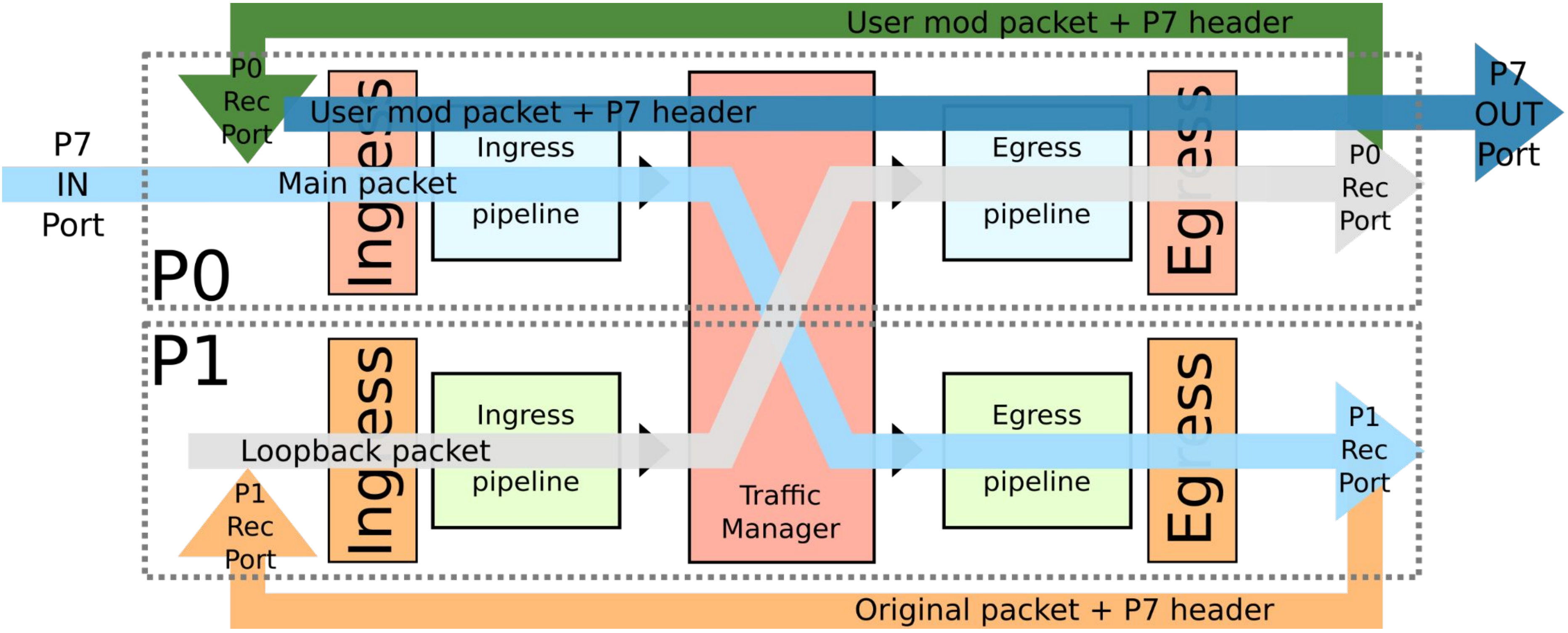
Architecture



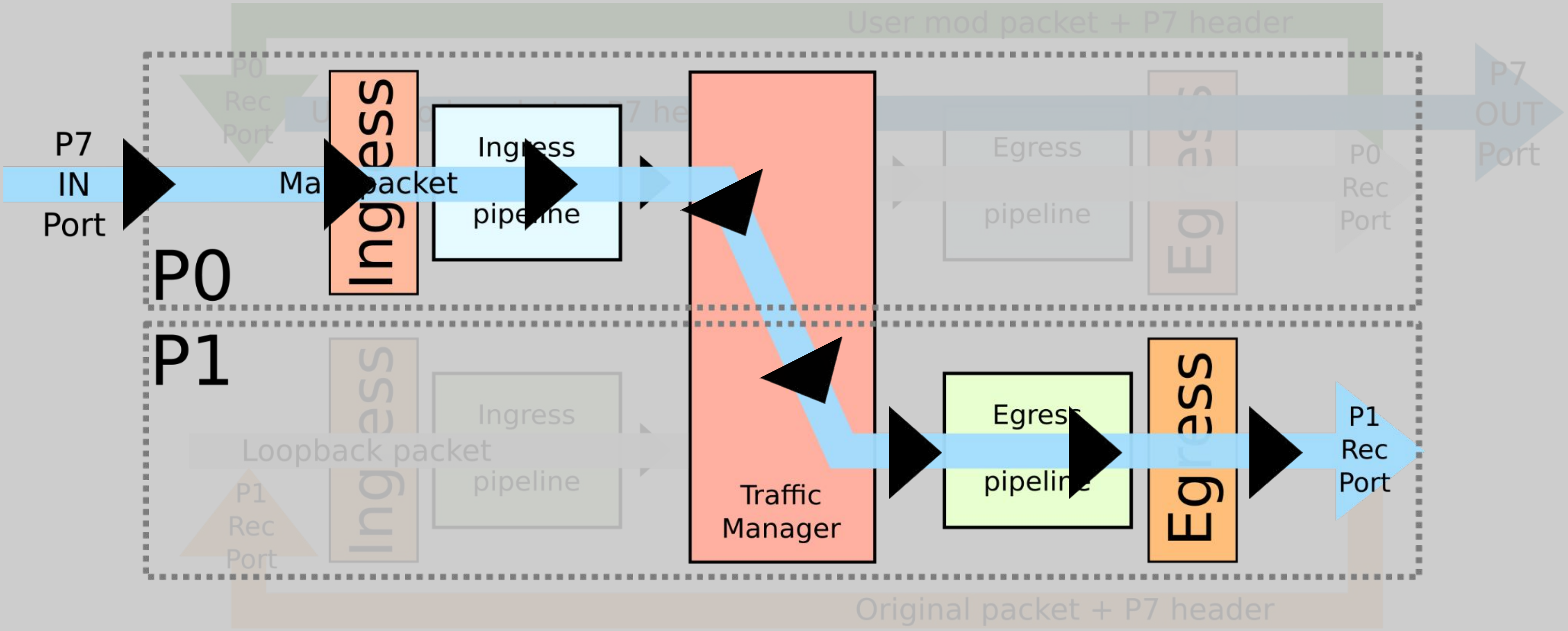
Architecture



Running P7

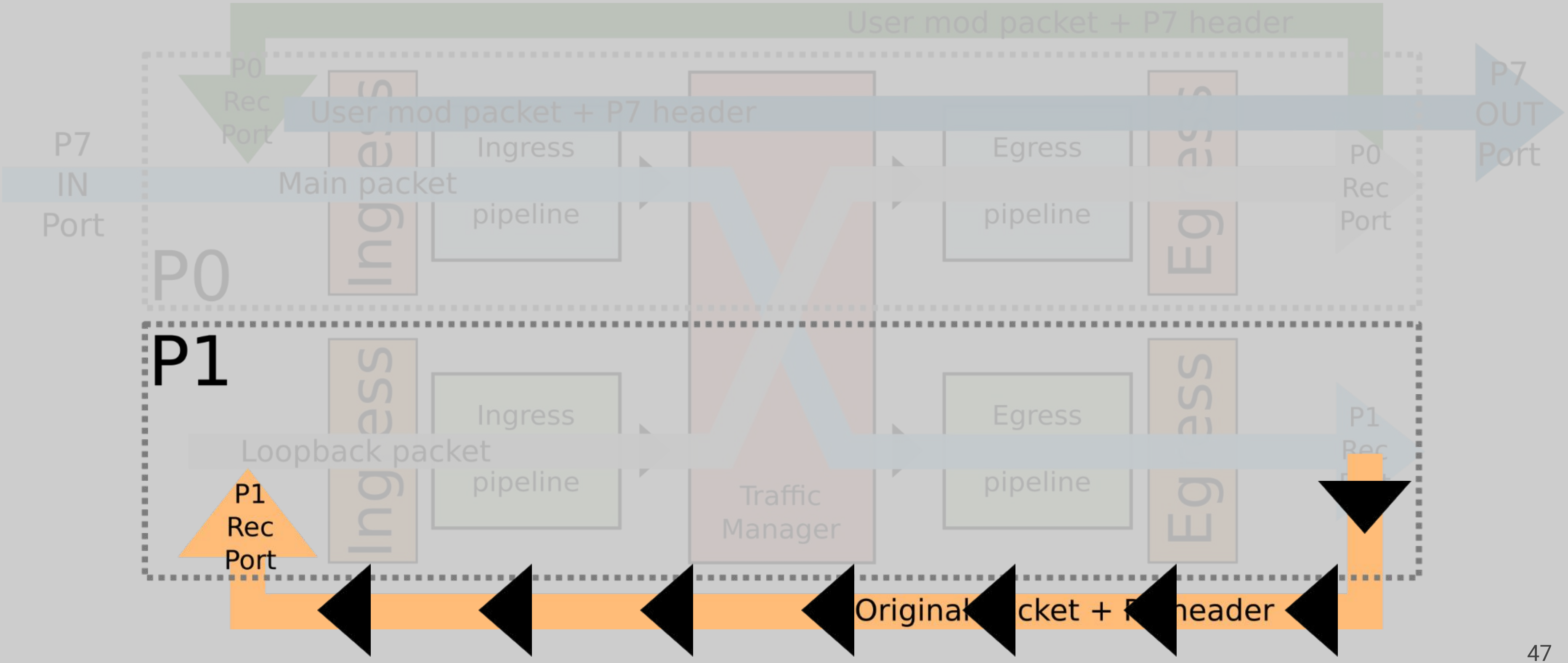

User mod packet + P7 header
P0
Rec
Port
User mod packet + P7 header
P7
OUT
Port
P7
IN
Port
Main packet
Ingress
Ingress
pipeline
Egress
pipeline
Egress
P0
Rec
Port
P0
P1
Ingress
Ingress
pipeline
Loopback packet
P1
Rec
Port
Traffic
Manager
Egress
pipeline
Egress
P1
Rec
Port
Original packet + P7 header

Running P7

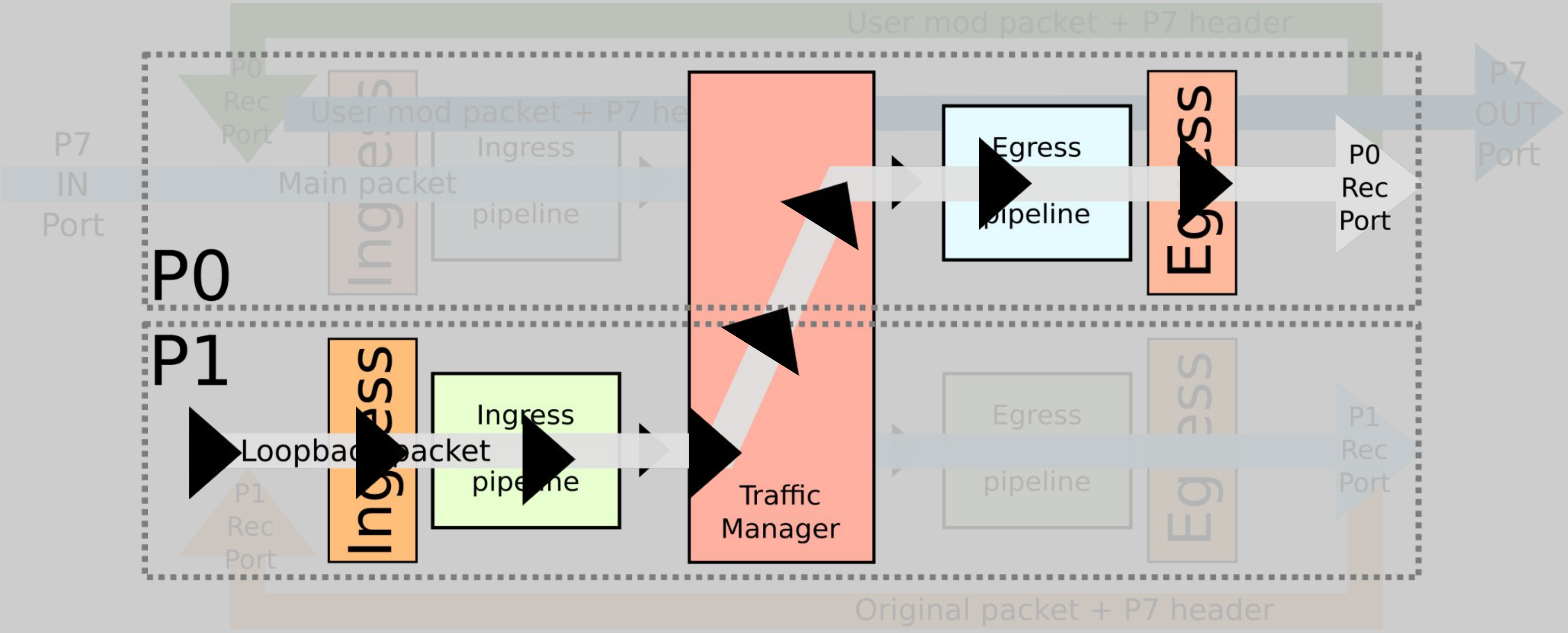


User mod packet + P7 header
P0 Rec Port
P7 OUT Port
P7 IN Port
Ingress
Ingress pipeline
Traffic Manager
Egress pipeline
Egress
P0 Rec Port
P0
P1
Ingress
Ingress pipeline
Loopback packet
P1 Rec Port
Egress pipeline
Egress
P1 Rec Port
Original packet + P7 header

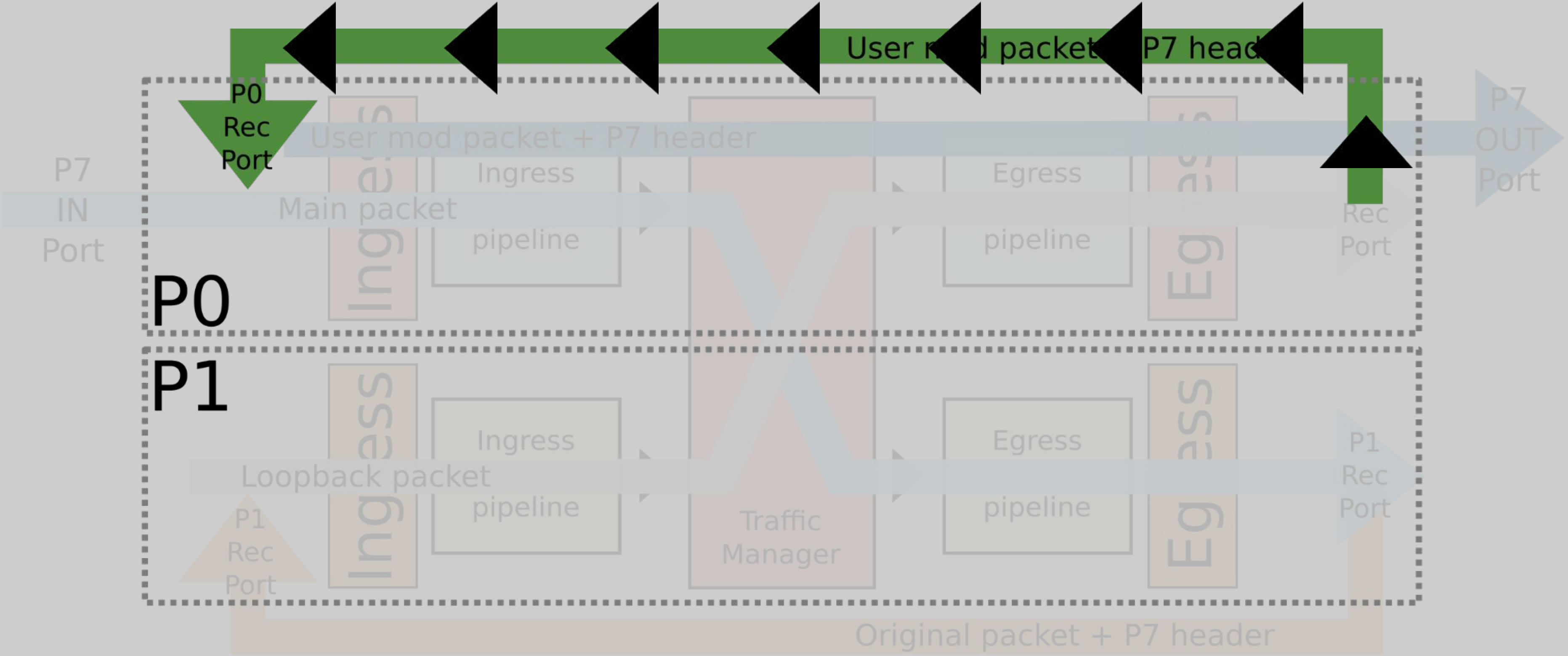
Running P7


User mod packet + P7 header
P0
Rec
Port
User mod packet + P7 header
P7
OUT
Port
P7
IN
Port
Main packet
Ingress
Ingress
pipeline
Egress
pipeline
Egress
P0
Rec
Port
P0
P1
Ingress
Ingress
pipeline
Traffic
Manager
Egress
pipeline
Egress
Loopback packet
P1
Rec
Port

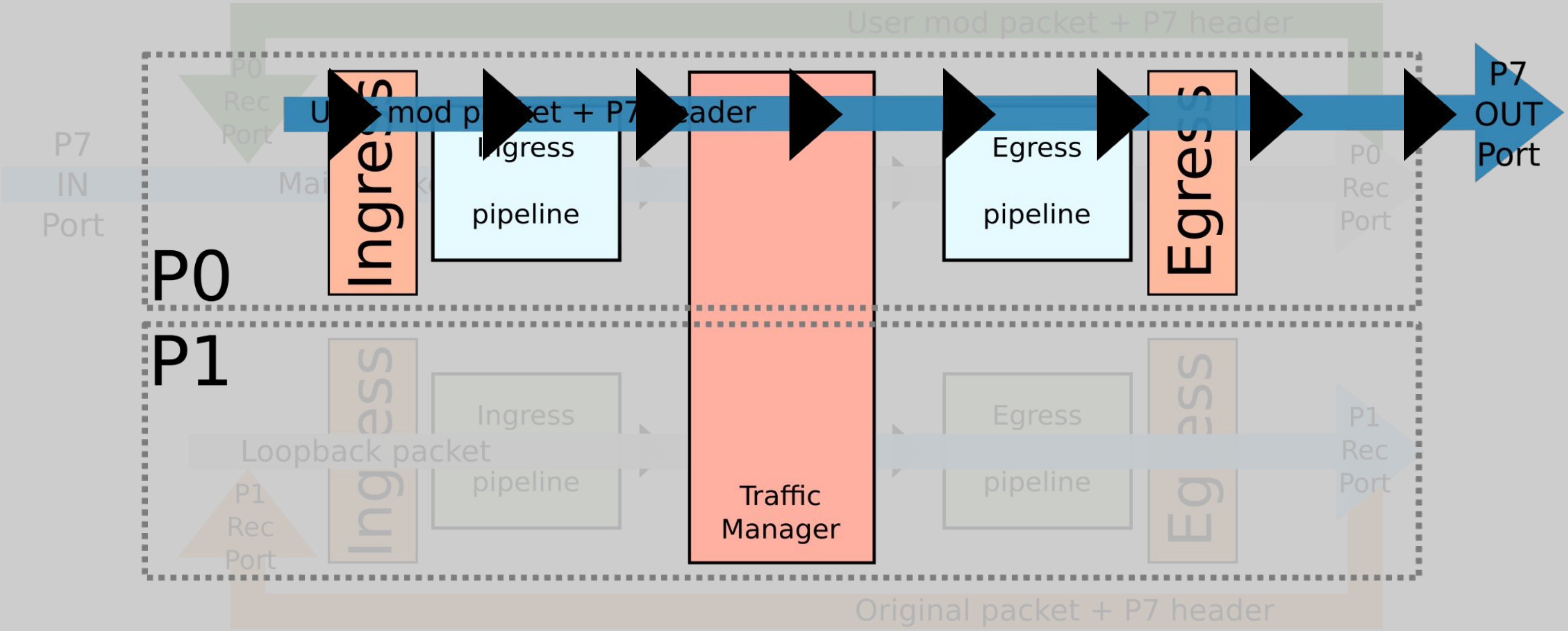
Running P7


User mod packet + P7 header
P0
Rec
Port
Ingress
User mod packet + P7 he
Ingress
pipeline
P7
IN
Port
Main packet
P0
Traffic
Manager
Egress
pipeline
Egress
P0
Rec
Port
P7
OUT
Port
P1
Ingress
Loopback packet
Ingress
pipeline
P1
Rec
Port
Egress
pipeline
Egress
P1
Rec
Port
Original packet + P7 header

Running P7

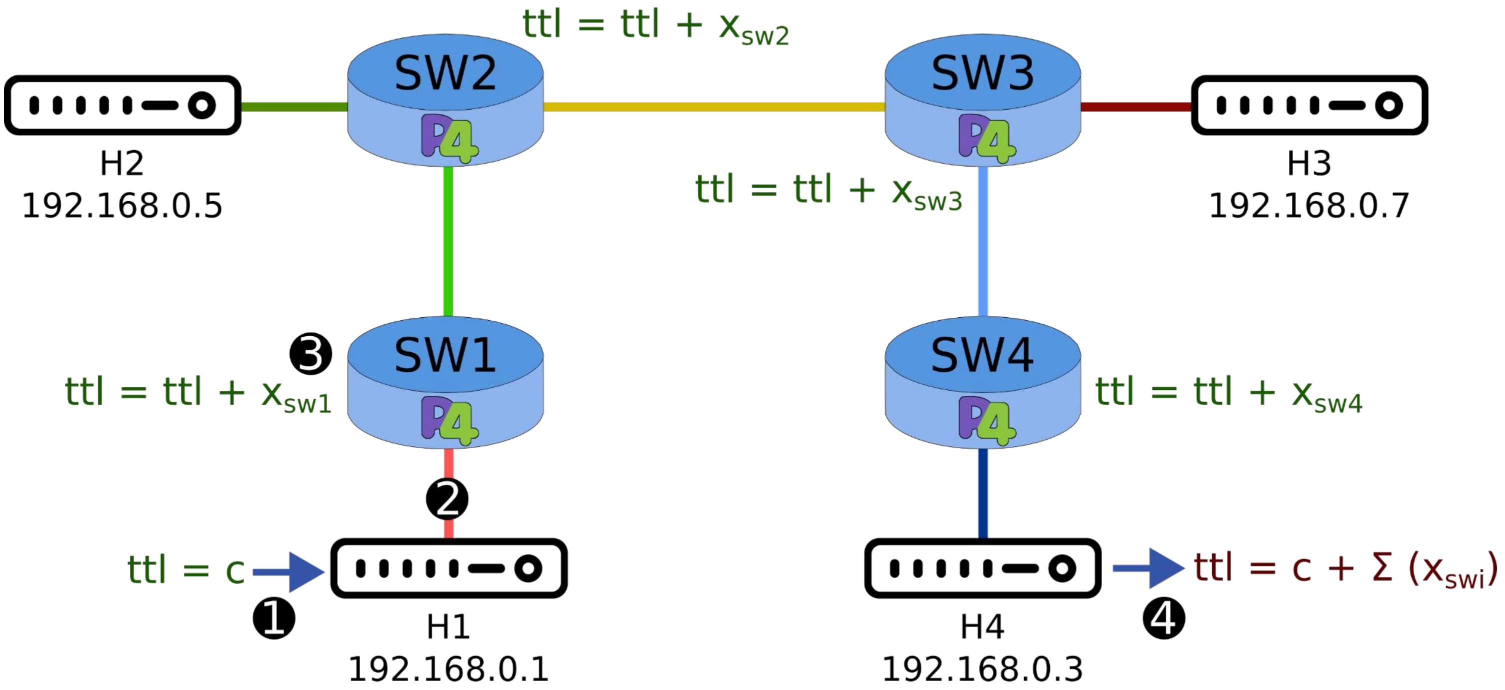

P0
Rec
Port
User mod packet + P7 header
P7
IN
Port
Main packet
Ingress
Ingress
pipeline
Egress
pipeline
Egress
P7
OUT
Port
Rec
Port
P0
P1
Ingress
Ingress
pipeline
Loopback packet
Traffic
Manager
Egress
pipeline
Egress
P1
Rec
Port
P1
Rec
Port
Original packet + P7 header

Running P7

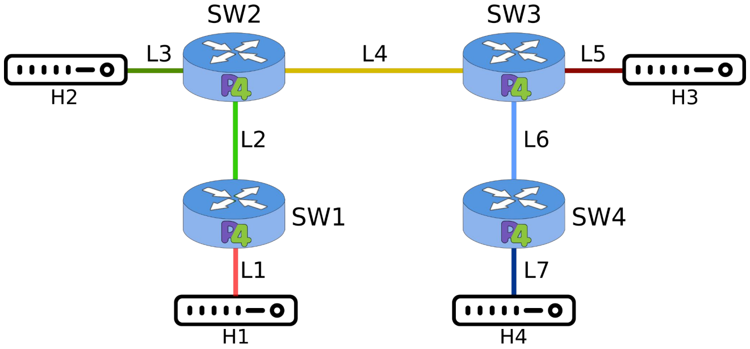


User mod packet + P7 header
P7 IN Port
P0 Rec Port
P0
Ingress
Ingress pipeline
Traffic Manager
Egress pipeline
Egress
P7 OUT Port
P1
Ingress
Ingress pipeline
Egress pipeline
Egress
P1 Rec Port
Loopback packet
P1 Rec Port
Original packet + P7 header

Multiple pipelines with P7

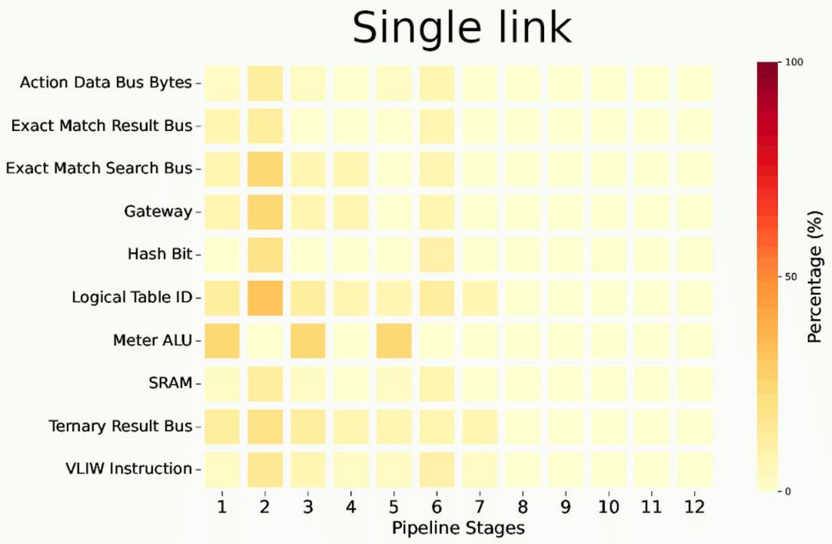


Multiple pipelines with P7



			Operation "+" Value				
			sw1	sw2	sw3	sw4	
Destination	192.168.0.1	h1	5	6	7	8	
	192.168.0.5	h2	15	16	17	18	
	192.168.0.7	h3	20	21	22	23	
	192.168.0.3	h4	10	11	12	13	
	In Value	**host 1**	**host 2**	**Result**			
	1	h1	h2	32	= 1 + 15 + 16		

Resource Utilization



Resource Utilization

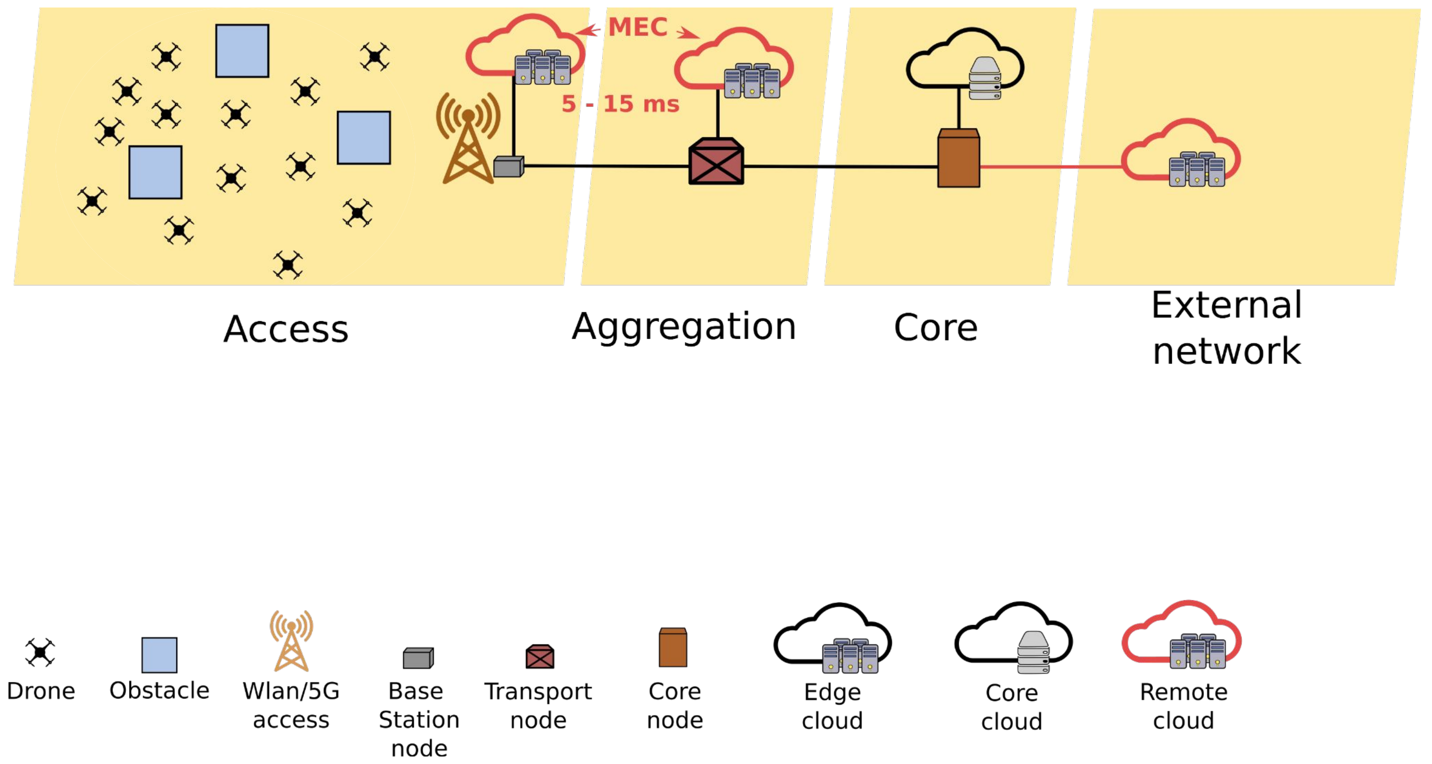
Resource	simple	single_sw	calculator	s_tree	5G	c_tree
Hash Bits	3.3%	3.3%	3.3%	3.3%	3.3%	3.3%
SRAM	4.5%	4.5%	4.5%	4.5%	4.5%	4.5%
TCAM	8.3%	8.3%	8.3%	8.3%	8.3%	8.3%
VLIW Actions	4.6%	5.9%	12.5%	16.6%	18.4%	27.8%
Gateway	4.7%	6.8%	17.2%	23.4%	25.5%	35.9%
Meter ALUs	6.3%	6.3%	6.3%	6.3%	6.3%	6.3%
Logical Table ID	8.3%	11.4%	27.0%	36.4%	39.5%	55.2%
Ternary Result Bus	6.2%	9.4%	25.0%	34.4%	37.5%	53.1%

04.

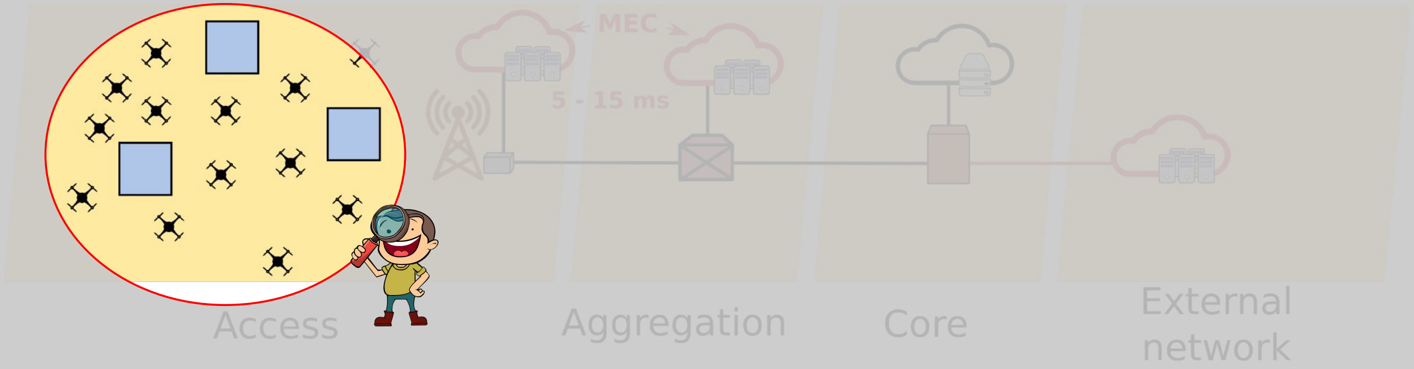
In-network Collision Avoidance

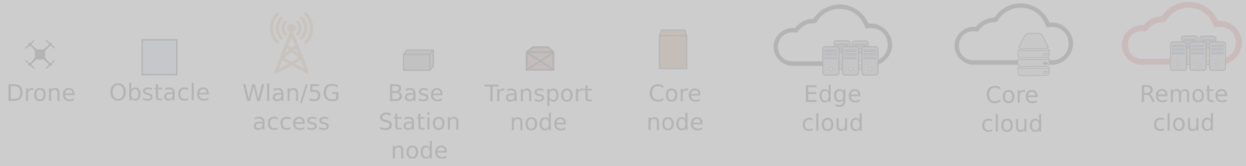
- Collision Avoidance Algorithms
- P4 approach

UAV Scenario

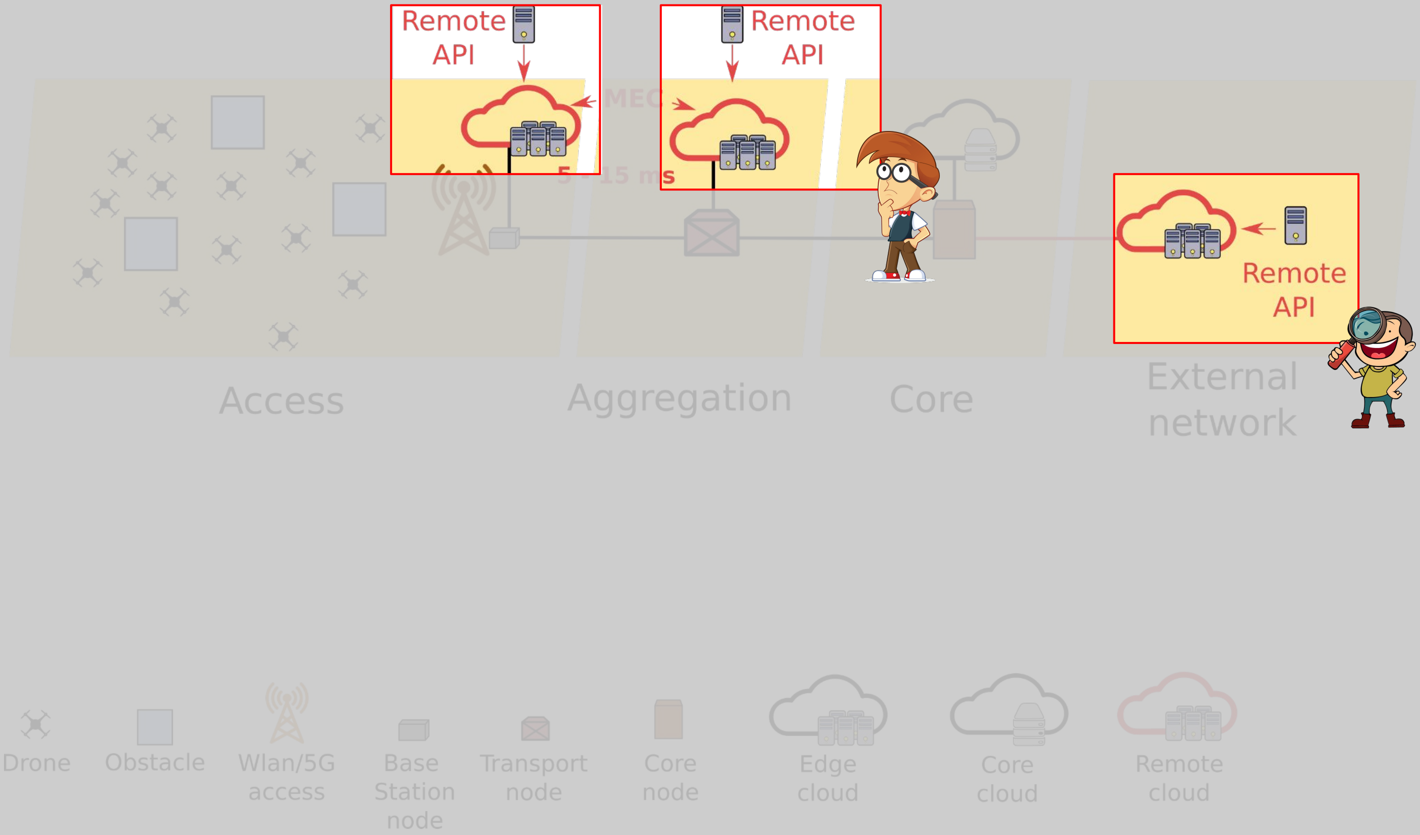


UAV Scenario

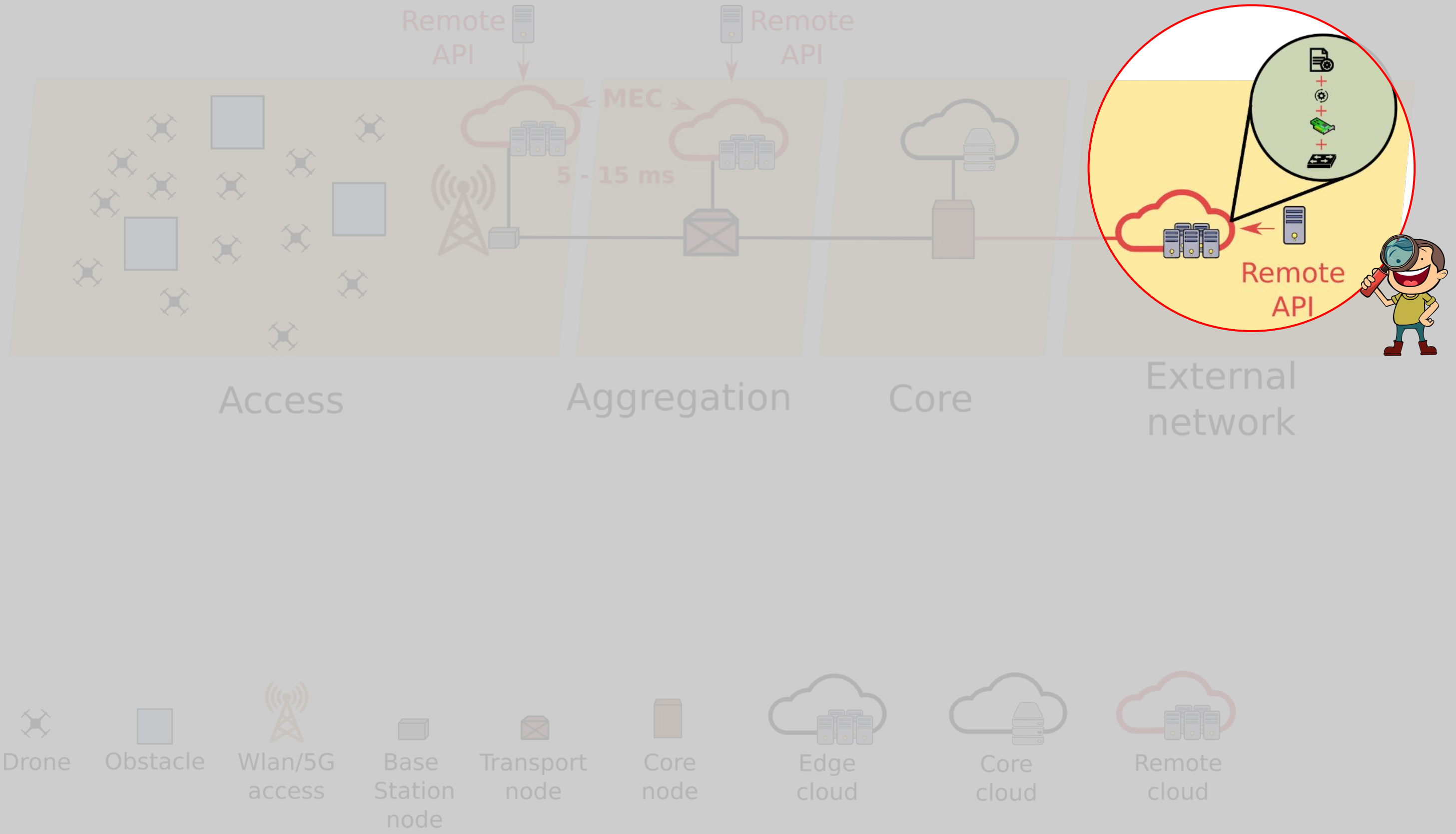




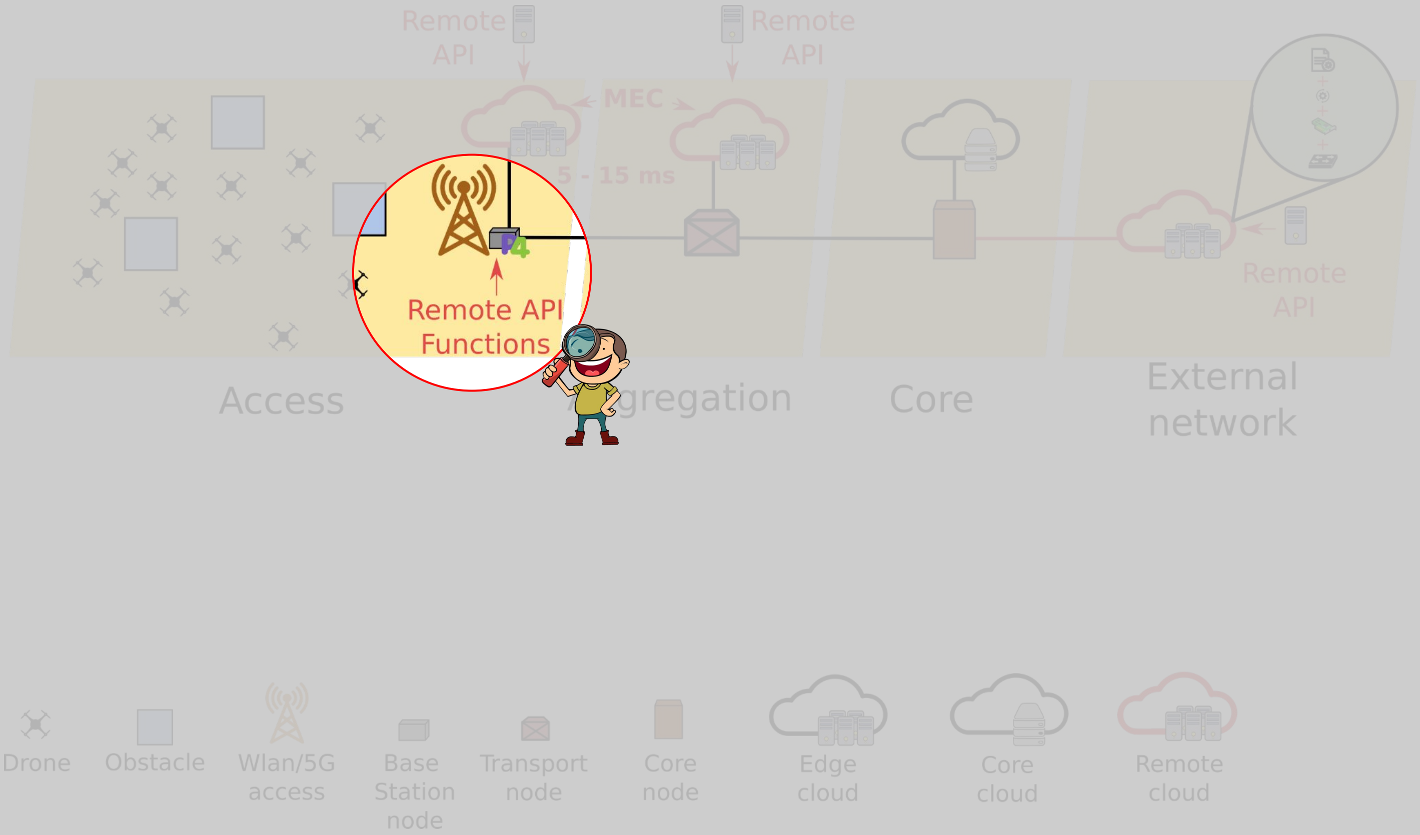
UAV Scenario



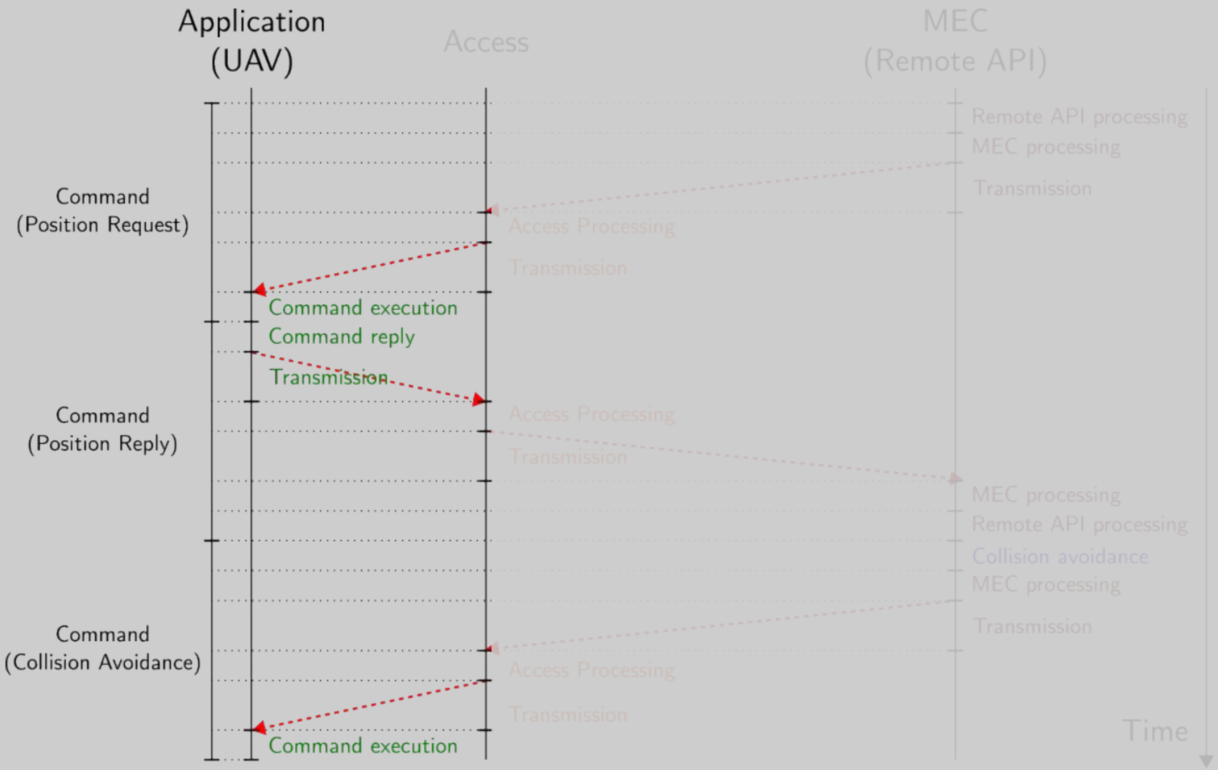
UAV Scenario



UAV Scenario



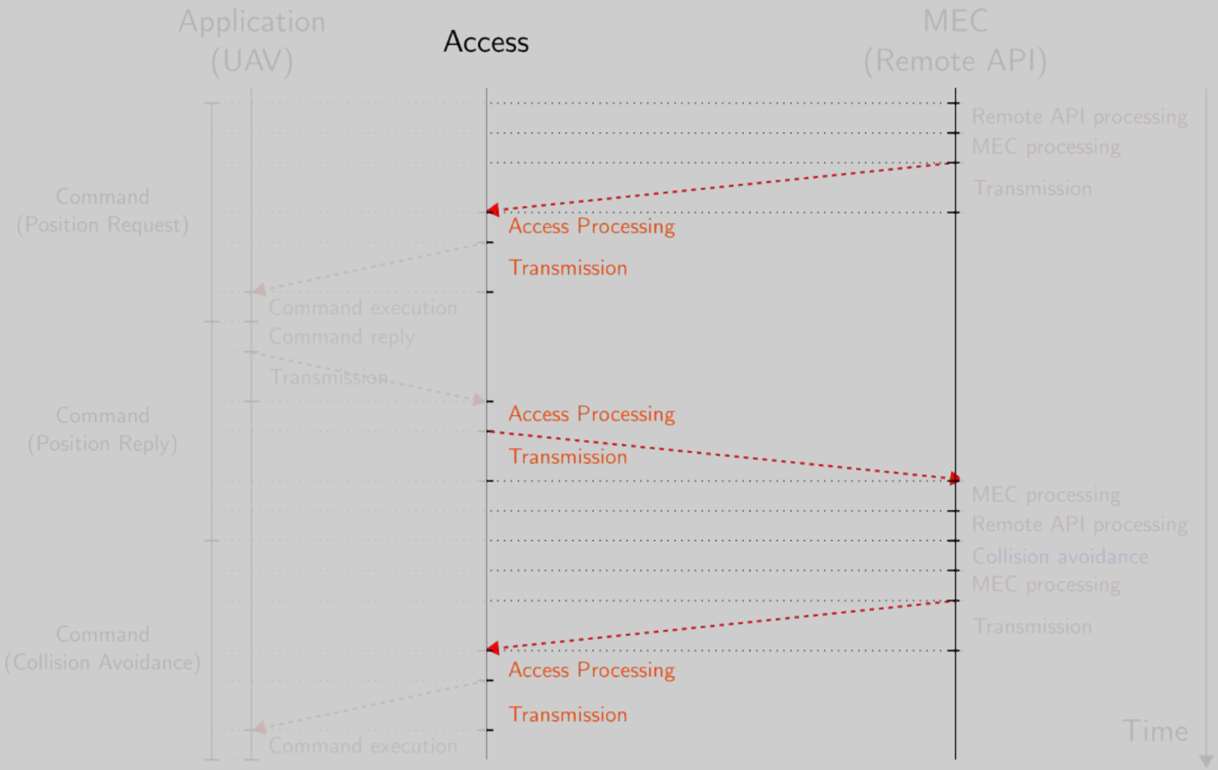
End-to-end application latency



Application, Access, MEC, 3DVFH+

$$Tt = \underbrace{0.05ms}_{Application} + \underbrace{2.2ms}_{Access} + \underbrace{5\ ms}_{MEC} + \underbrace{0.3ms}_{3DVFH+}$$

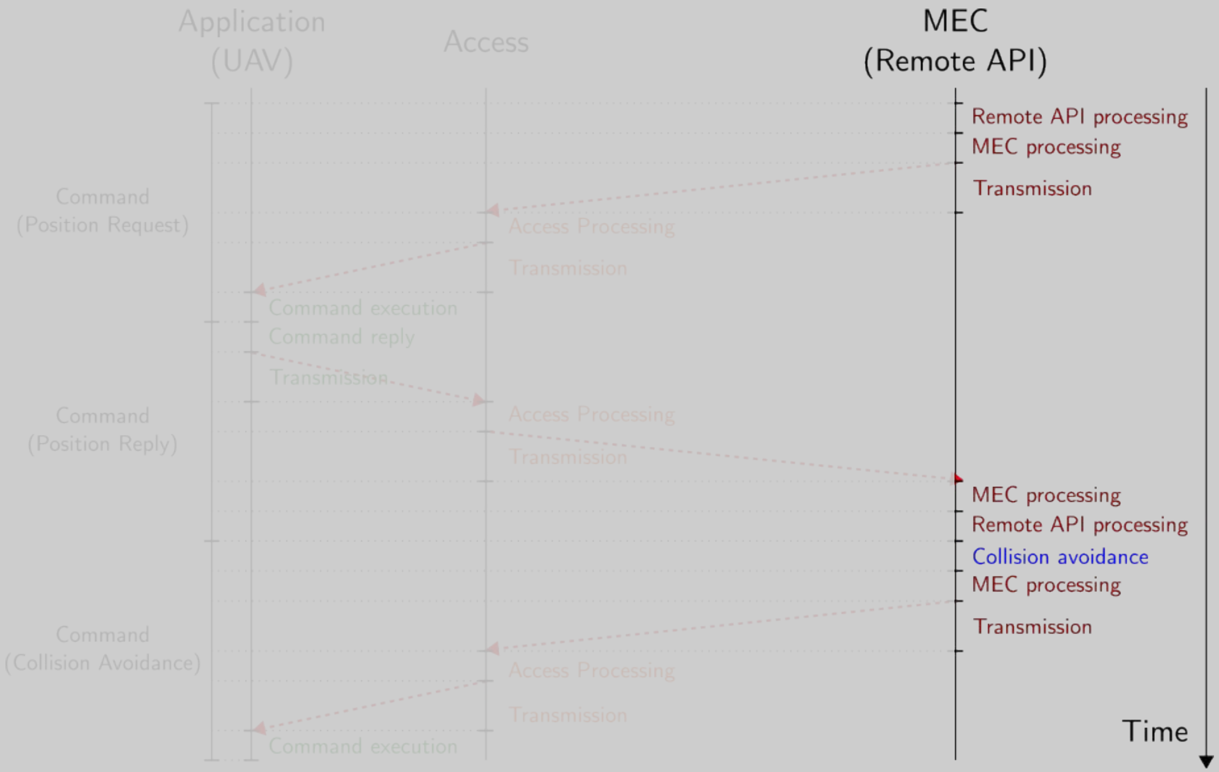
End-to-end application latency



Application Access MEC 3DVFH+

Tt = 0.05ms + 2.2ms + 5 ms + 0.3ms

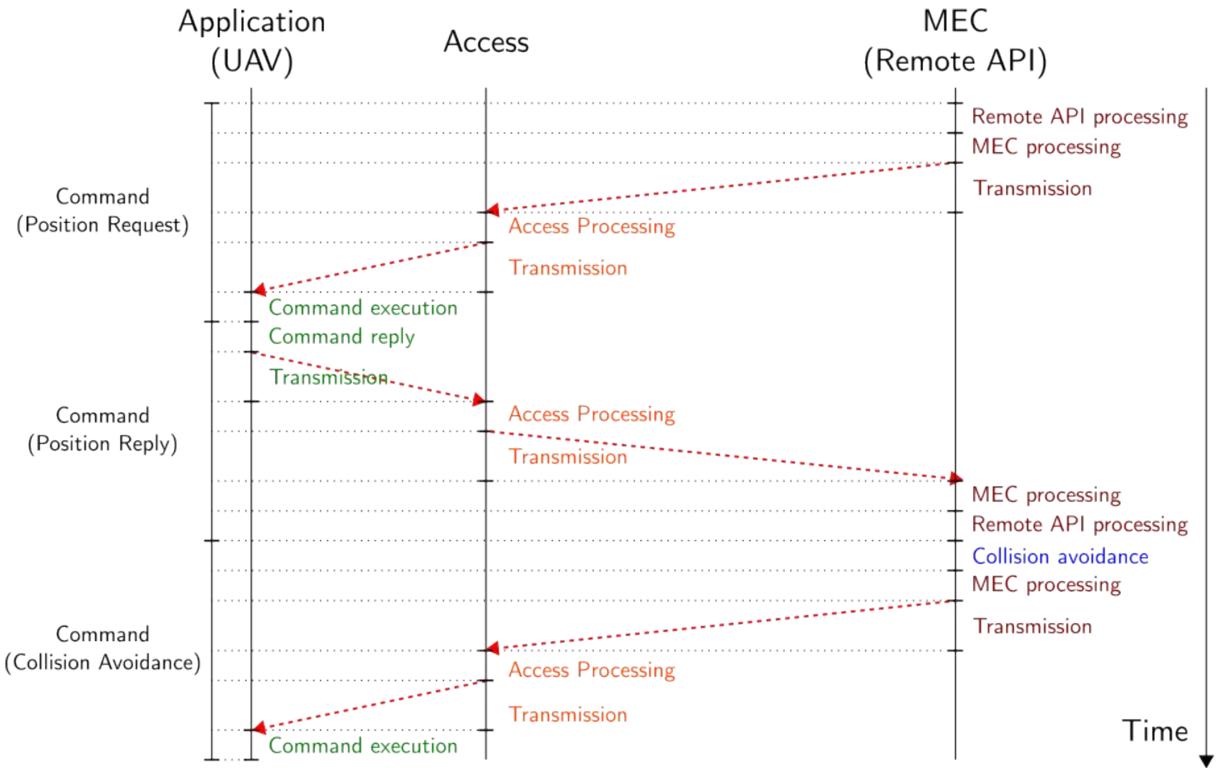
End-to-end application latency



Application, Access, MEC, 3DVFH+

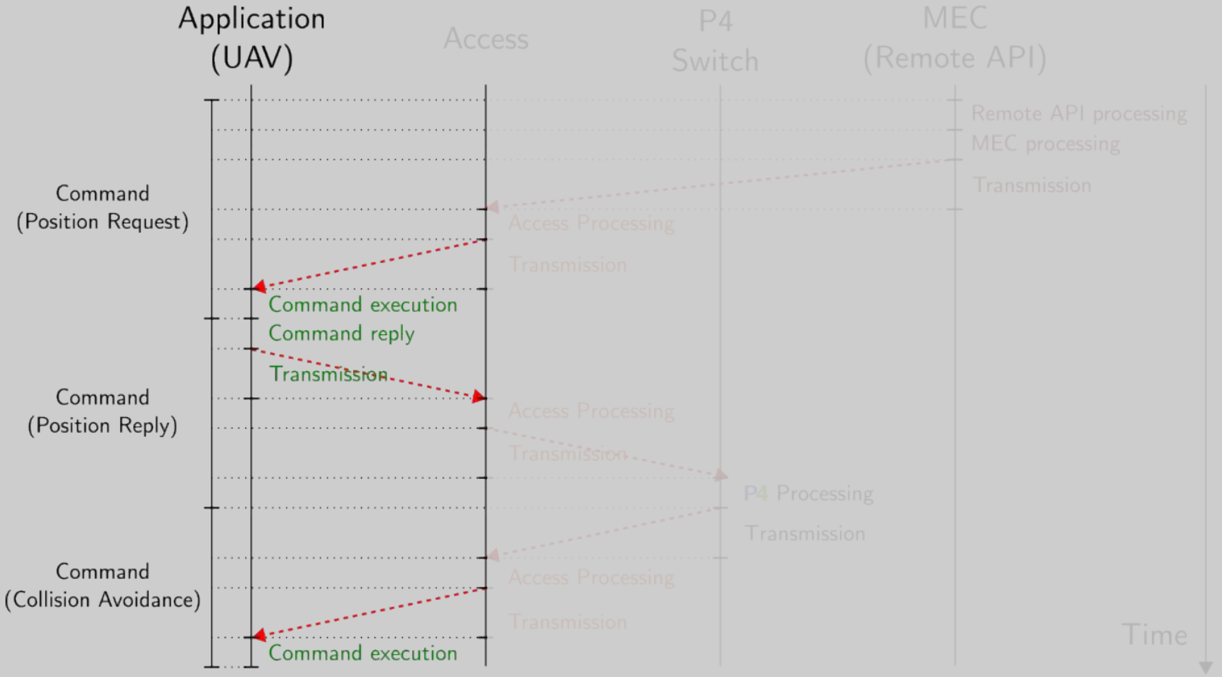
Tt = 0.05ms + 2.2ms + 5 ms + 0.3ms

End-to-end application latency



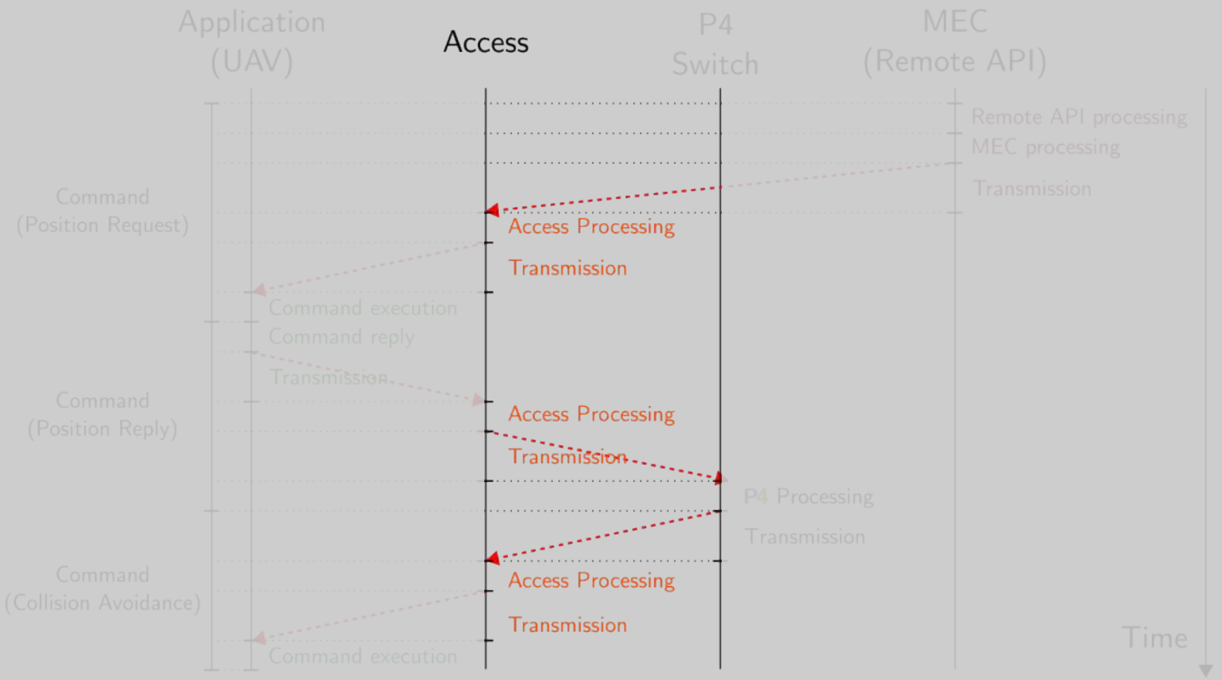
$$Tt = \overbrace{0.05ms}^{\text{Application}} + \overbrace{2.2ms}^{\text{Access}} + \overbrace{5\ ms}^{\text{MEC}} + \overbrace{0.3ms}^{\text{3DVFH+}}$$

End-to-end application latency



$$Tt_{P4} = \underbrace{0.05ms}_{\text{Application}} + \underbrace{2.2ms}_{\text{Access}} + \underbrace{P4}_{\text{In-network}}$$

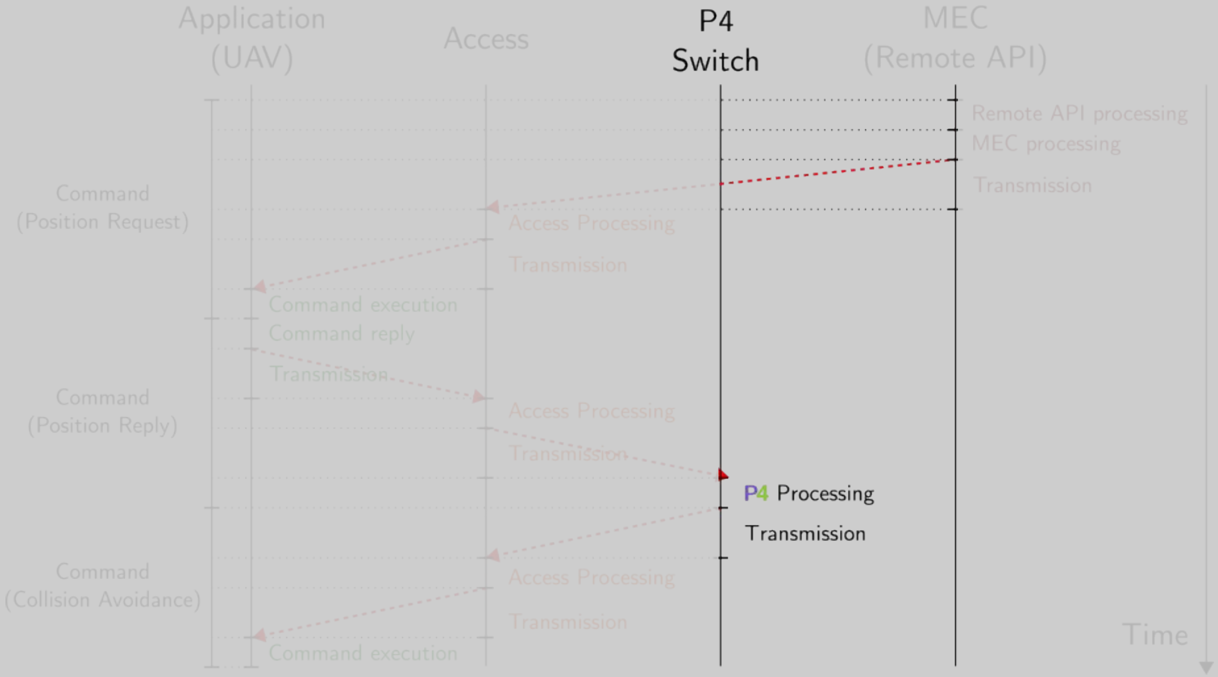
End-to-end application latency



Application Access In-network

$$Tt_{P4} = \underbrace{0.05ms}_{Application} + \underbrace{2.2ms}_{Access} + \underbrace{P4}_{In\text{-}network}$$

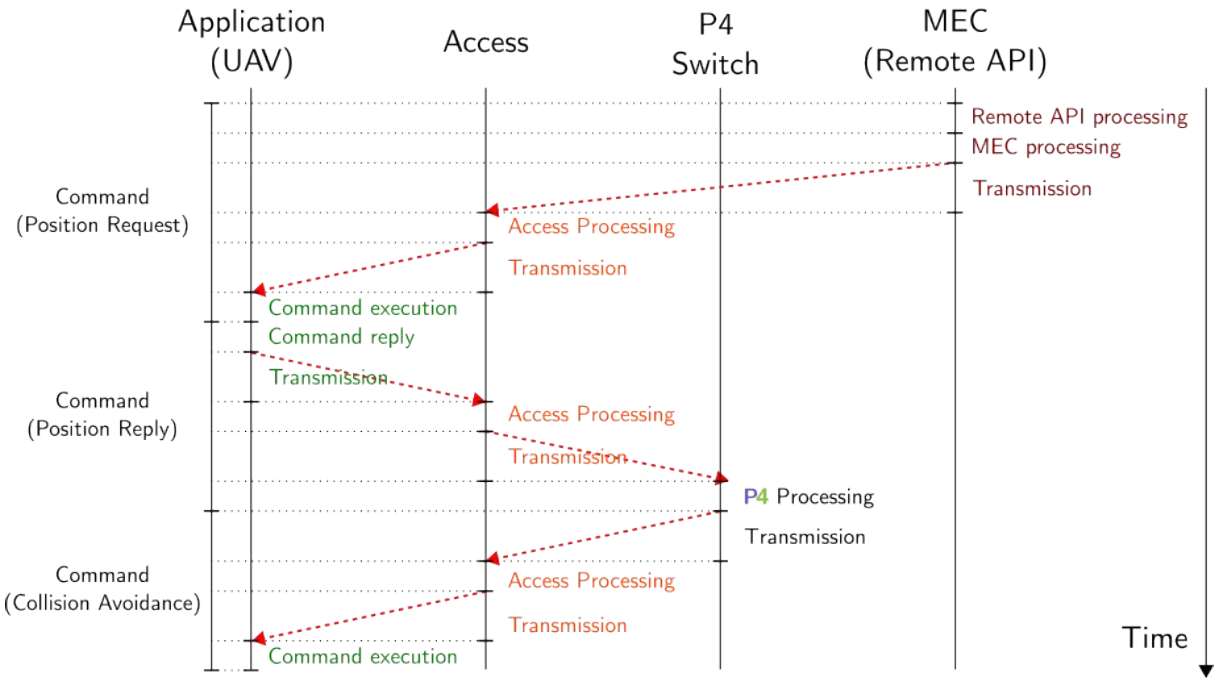
End-to-end application latency



Application Access In-network

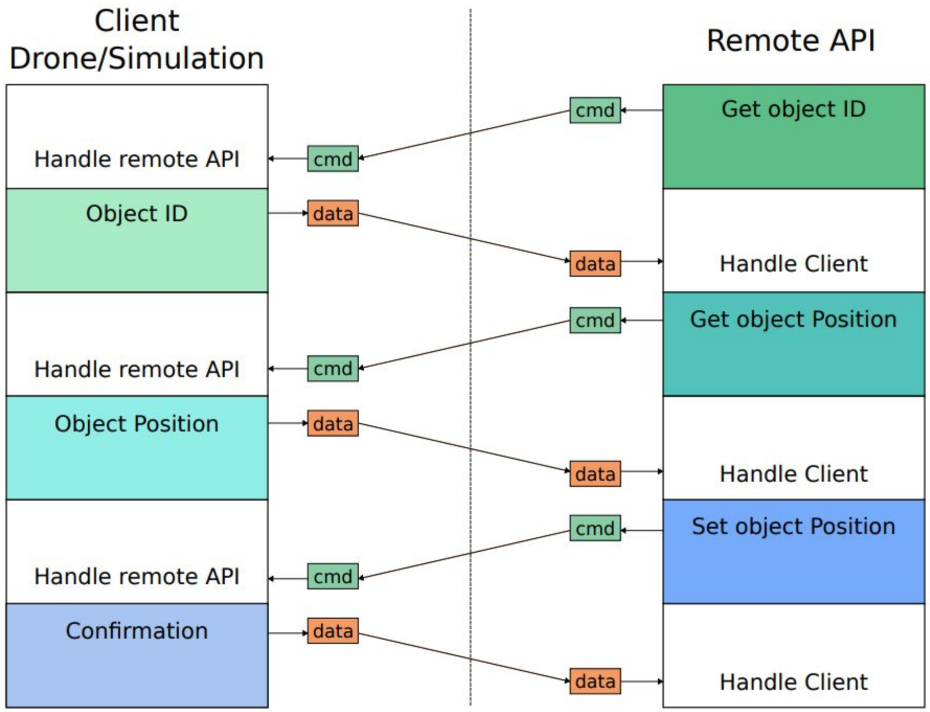
Tt_{P4} = 0.05ms + 2.2ms + P4

End-to-end application latency

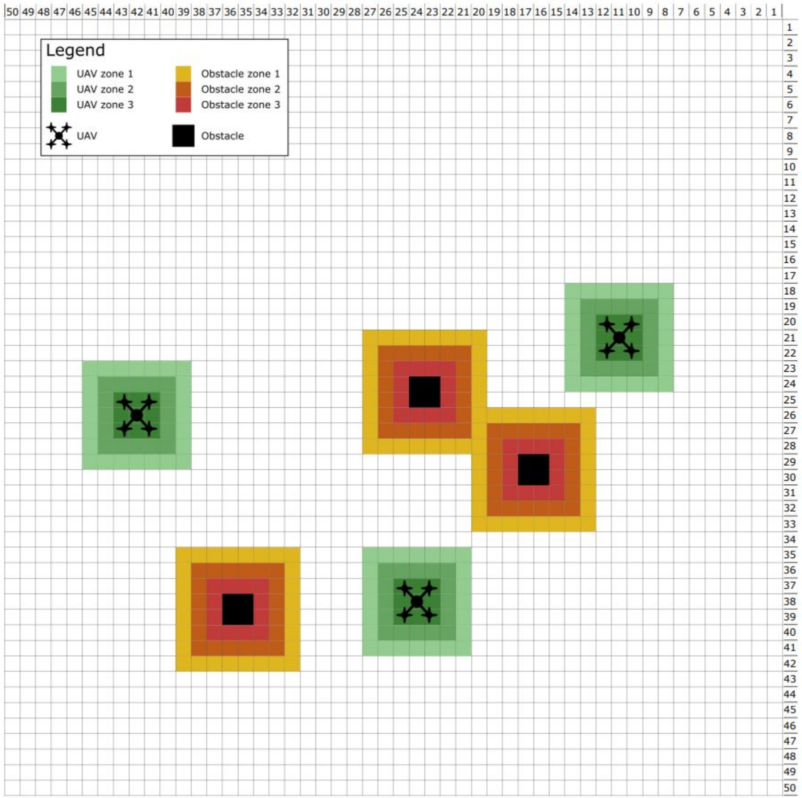


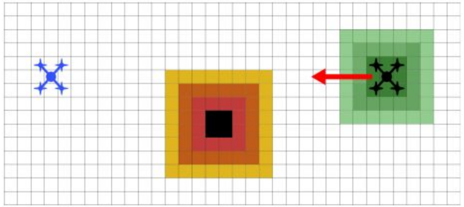
$$Tt_{P4} = \underbrace{0.05ms}_{\text{Application}} + \underbrace{2.2ms}_{\text{Access}} + \underbrace{P4}_{\text{In-network}}$$

UAV ⇒ Remote API

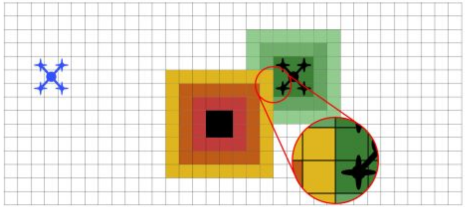


Histogram implementation and avoidance process

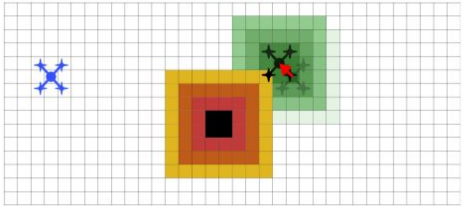




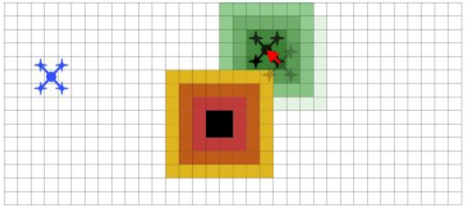
(a) Starting position.



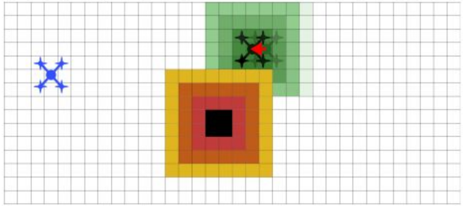
(b) Collision detection.



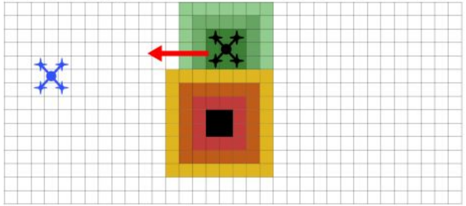
(c) Initial avoidance move.



(d) Avoidance movement.

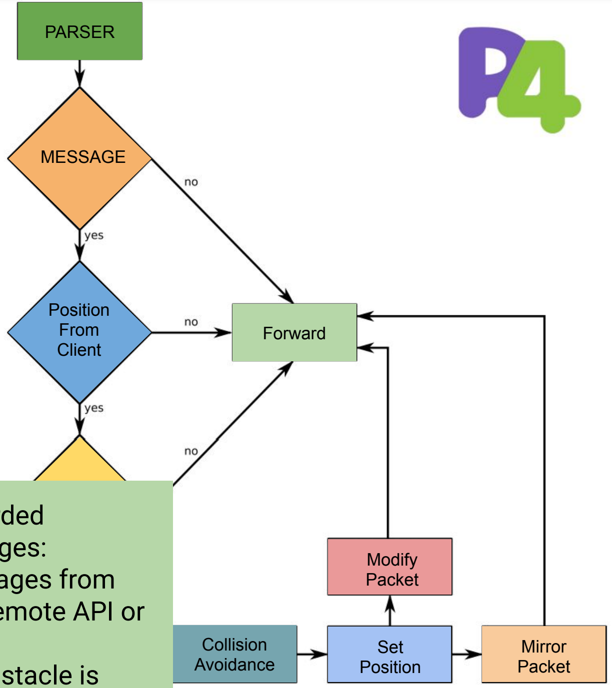


(e) New path.



(f) Following new path.

Avoidance algorithm in P4



Forwarded messages:

- Messages from the Remote API or client.
- No obstacle is detected.
- Modified packets.
- Mirrored packets.

Extract the packet headers of the messages.

It is verified the type of message and its content.

It is verified if the message is a get position Reply from the Client.

It is verified if there is an obstacle nearby.

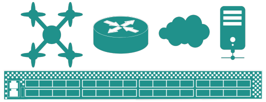
The Histogram is Updated using the information of the packet.

If an obstacle is detected, the avoidance process starts.

A new position is calculated to avoid the collision.

A mirror packet (clone of the original packet) is sent to the remote API

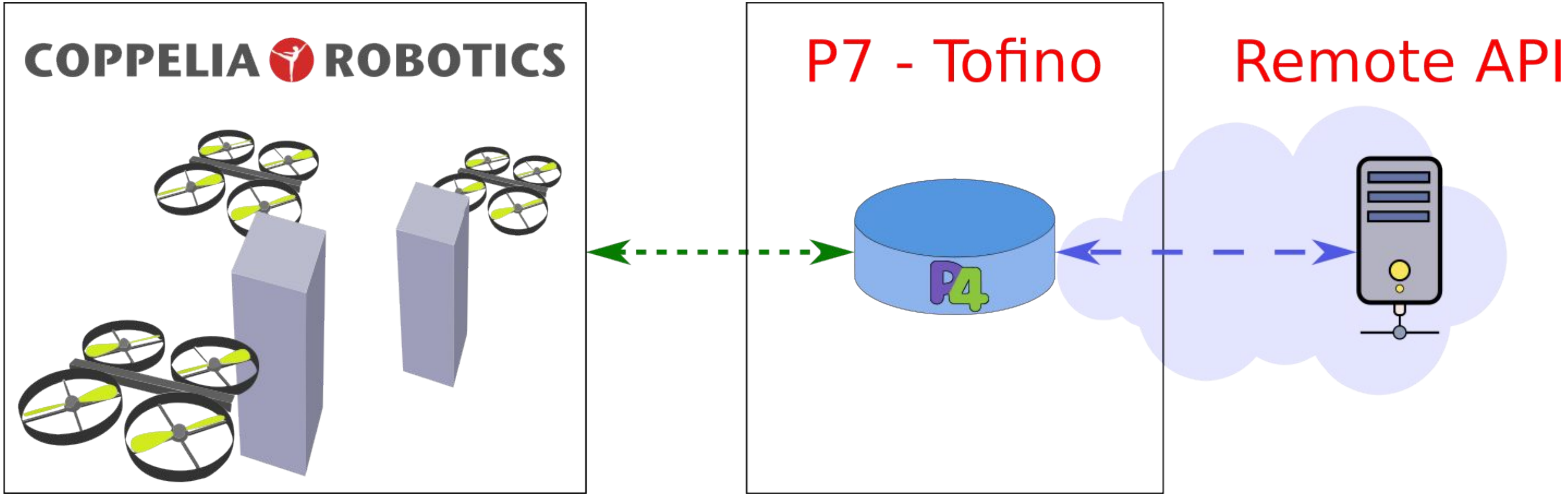
The original packet is updated with the new UAV’s position.

05. 

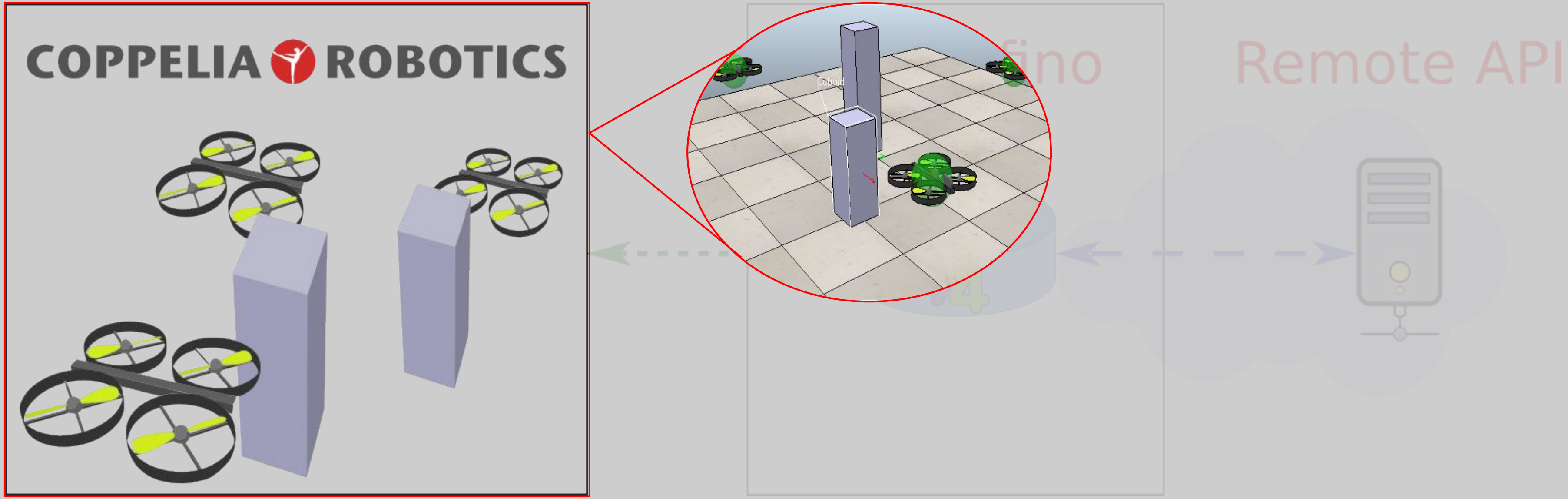
P7 + UAV

- P4 Collision Avoidance Algorithms over P7

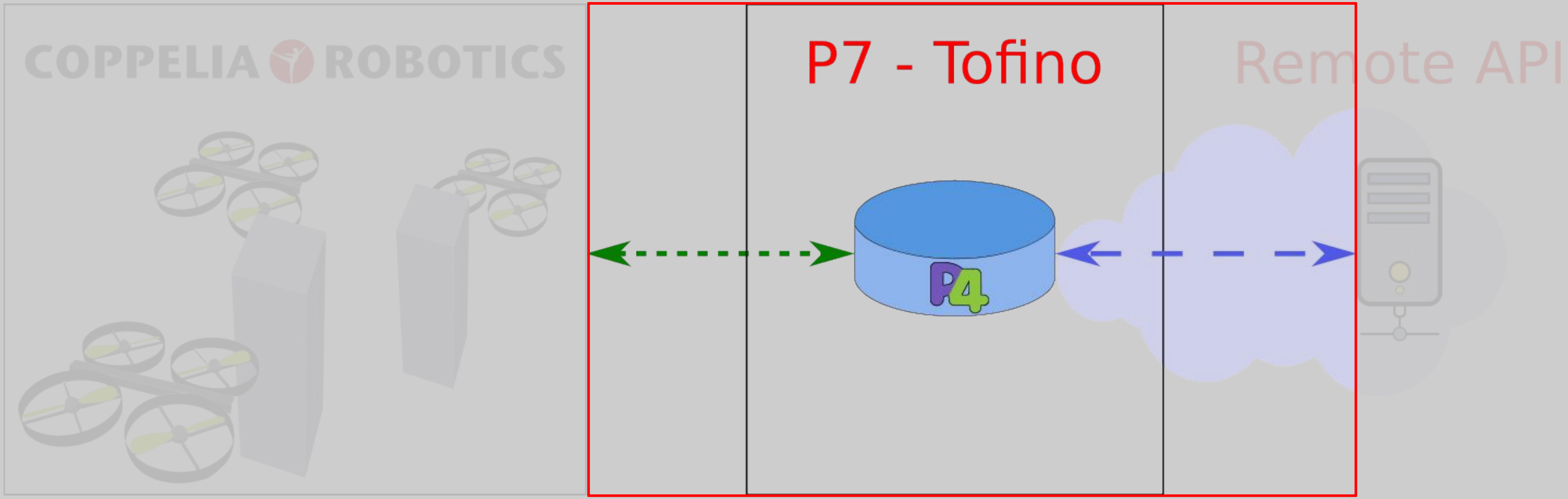
Collision Avoidance Algorithm + P7



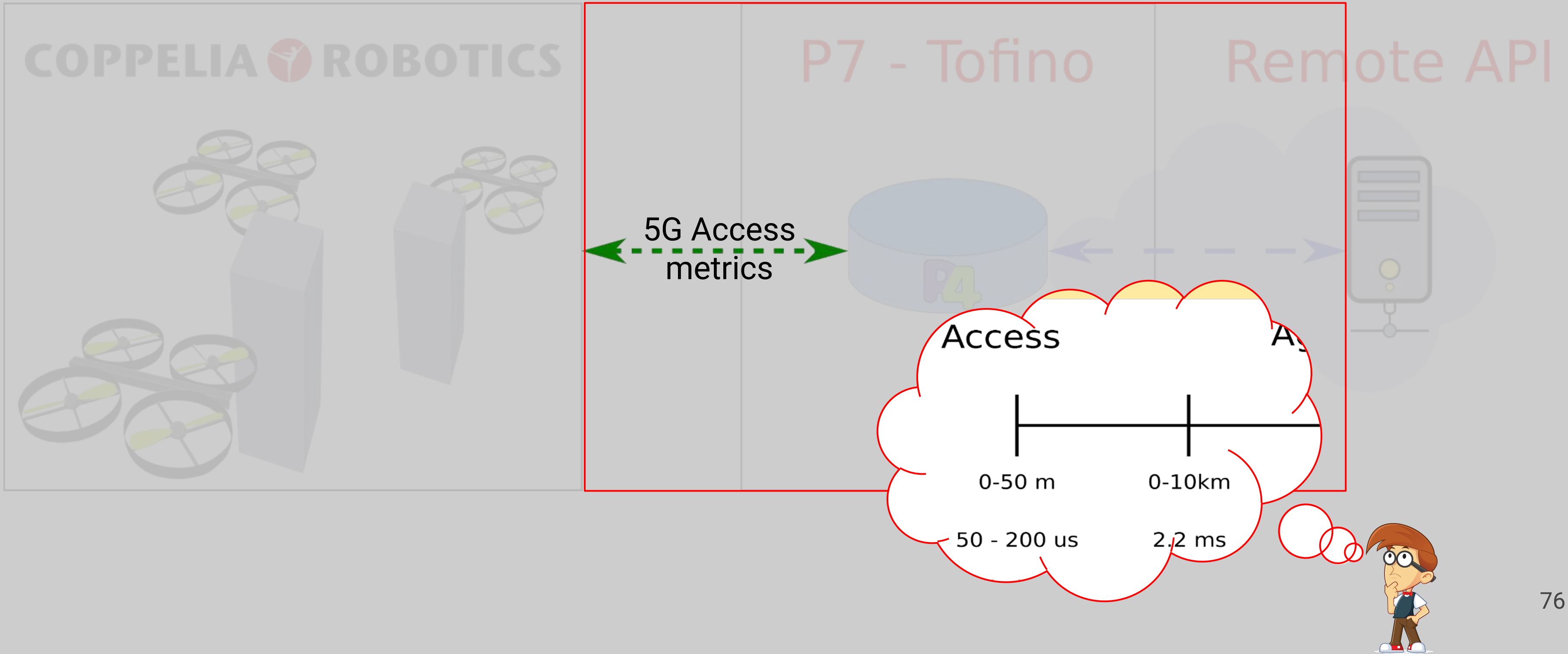
Collision Avoidance Algorithm + P7



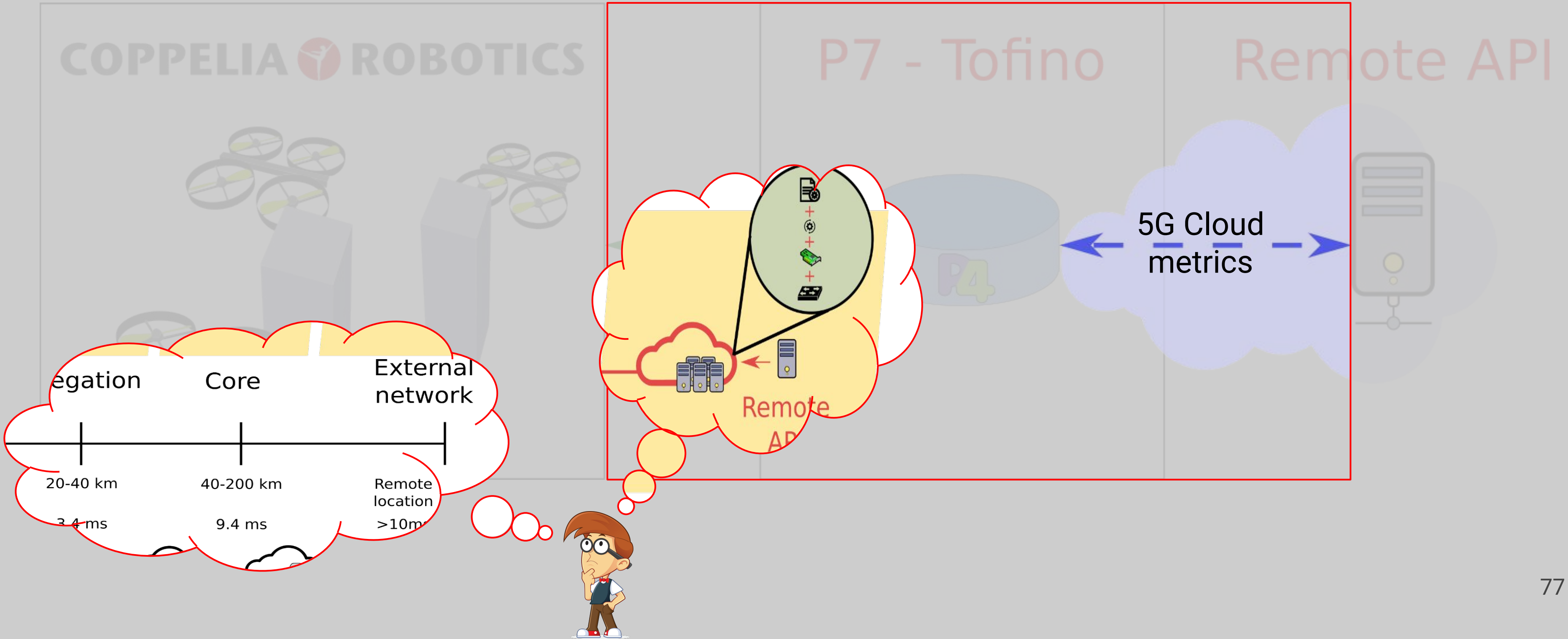
Collision Avoidance Algorithm + P7



Collision Avoidance Algorithm + P7



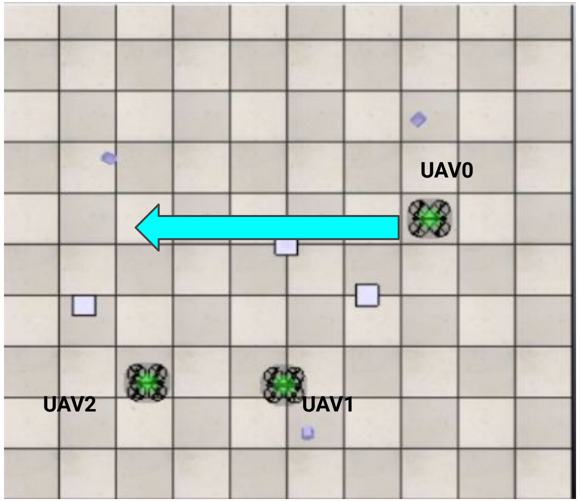
Collision Avoidance Algorithm + P7

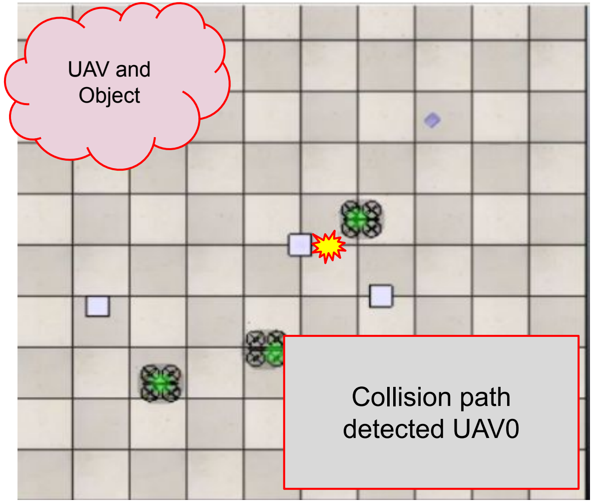


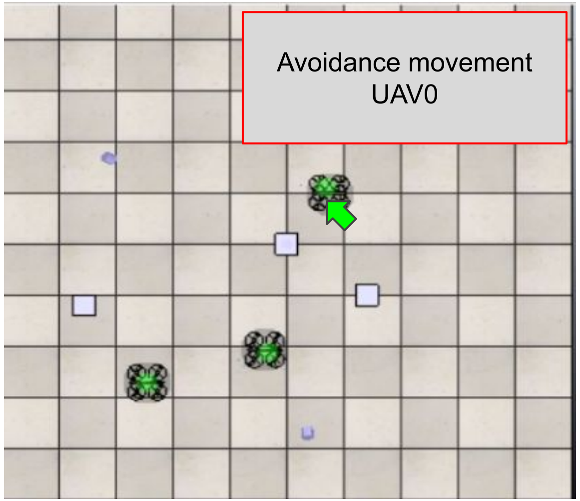
Collision Avoidance Algorithm + P7

Collision Avoidance

Initial position and destination.

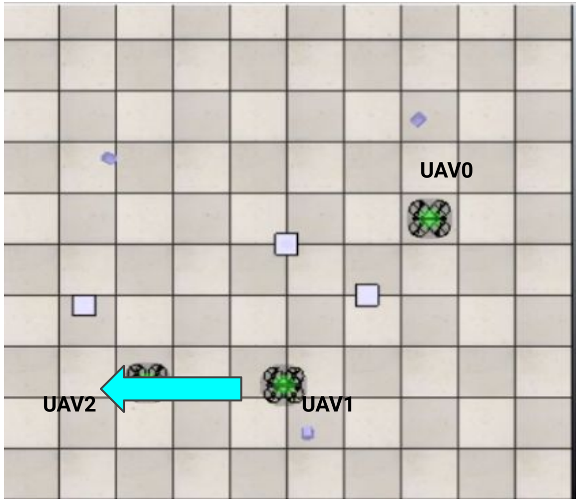


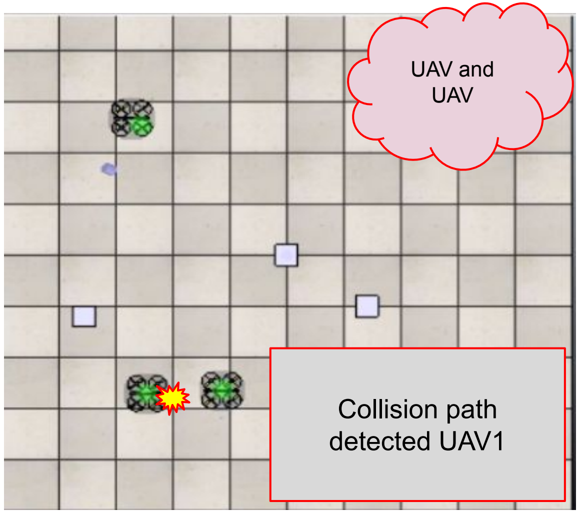


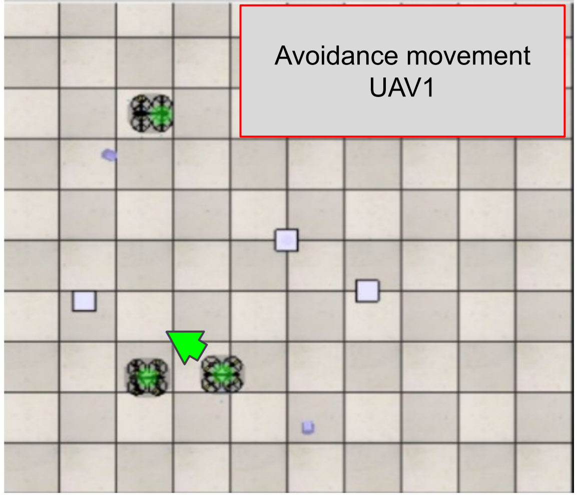


Collision Avoidance

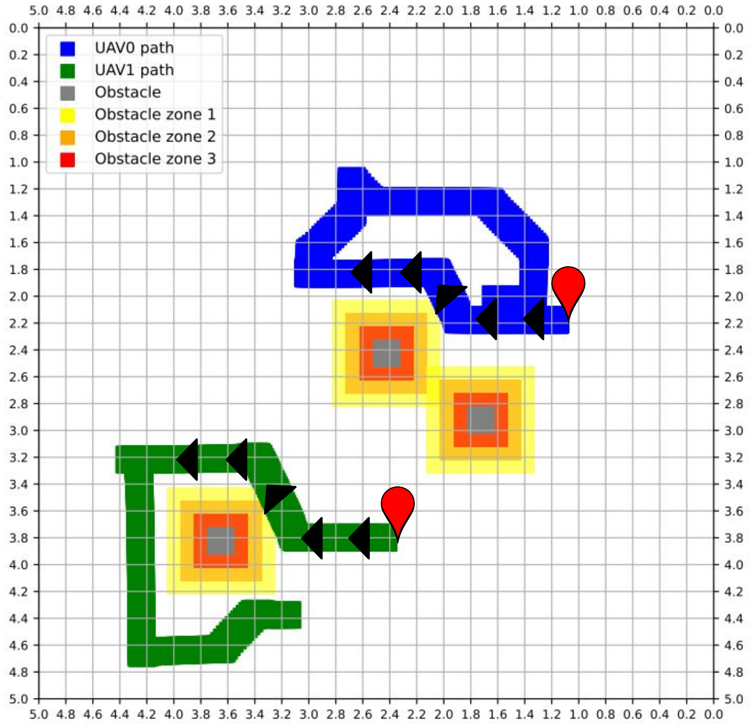
Initial position and destination.

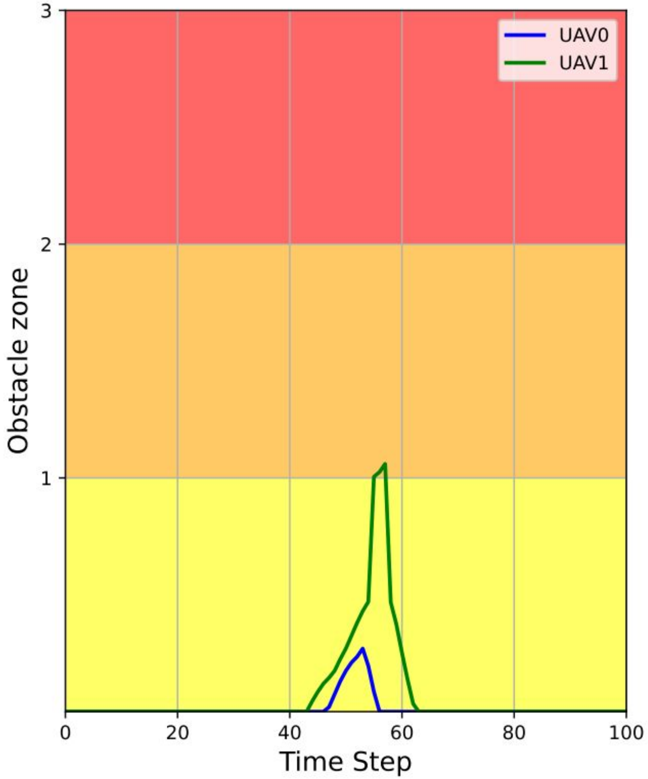




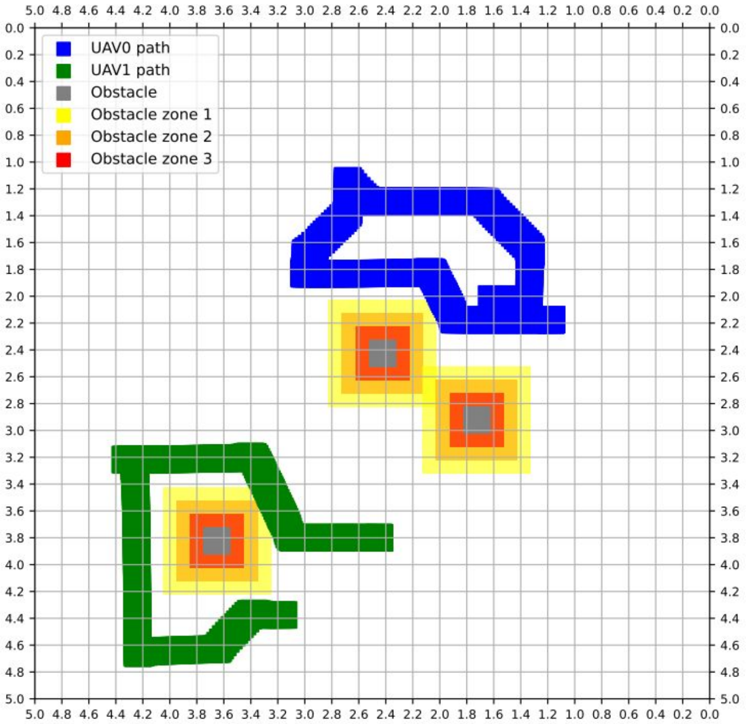


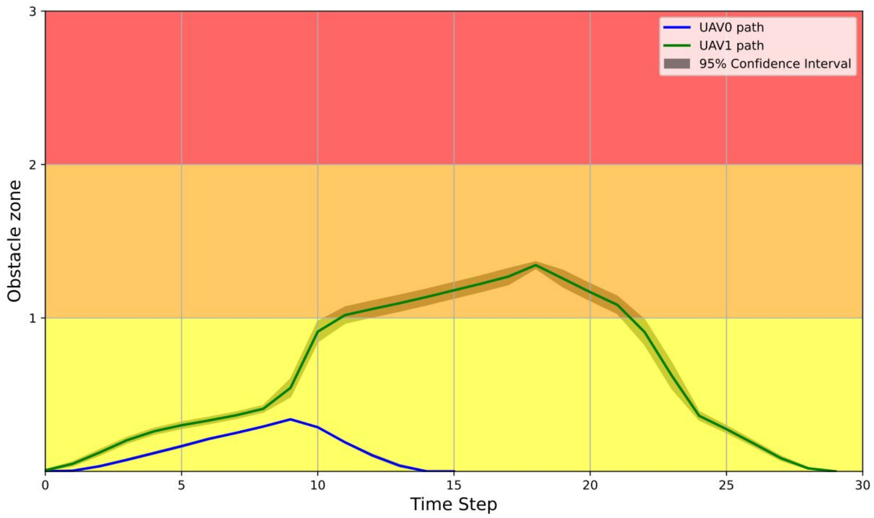
In-network collision avoidance scenario



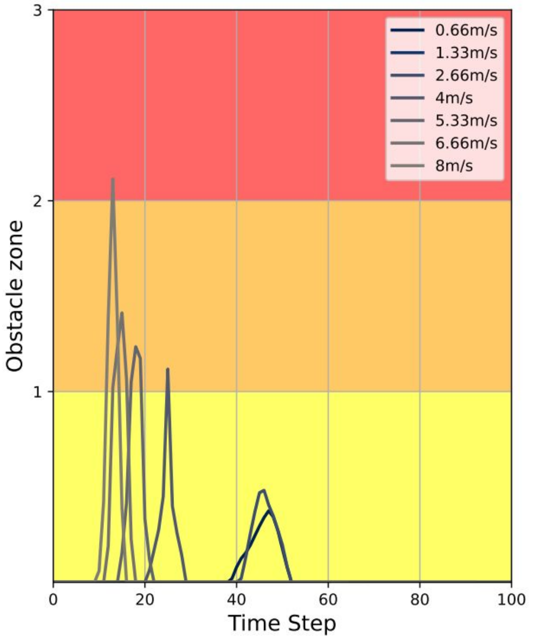


In-network collision avoidance scenario

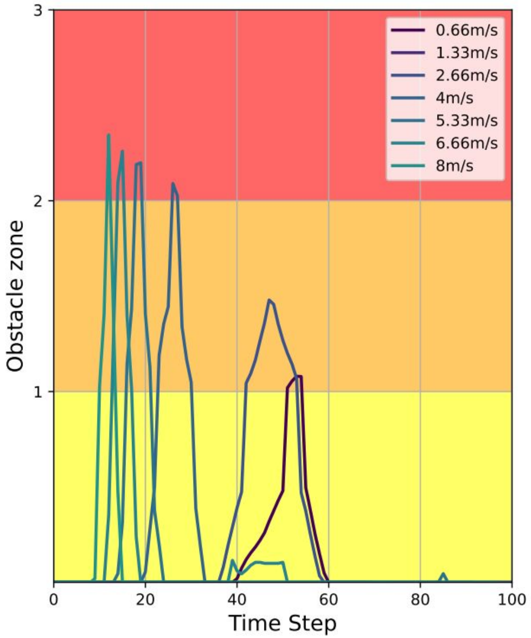




In-network collision avoidance scenario

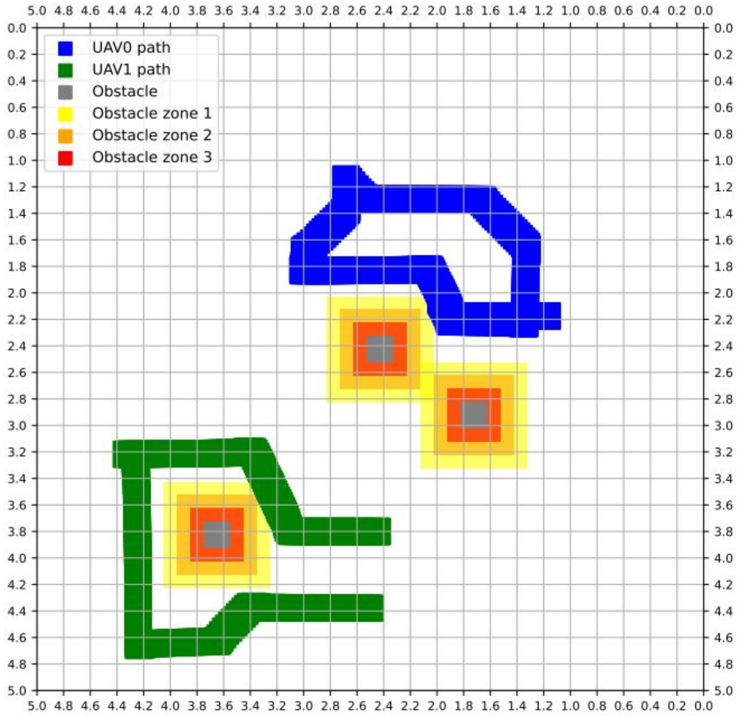


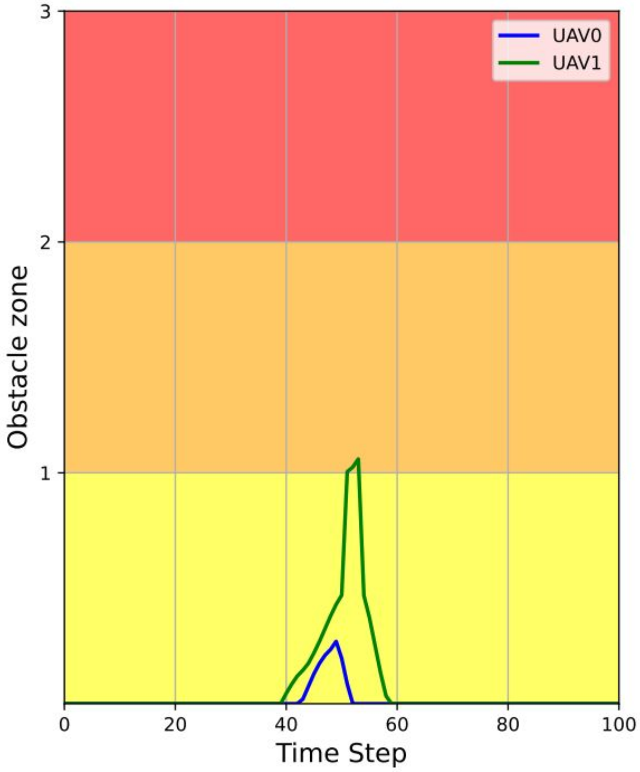
(a) UAV 0 path.



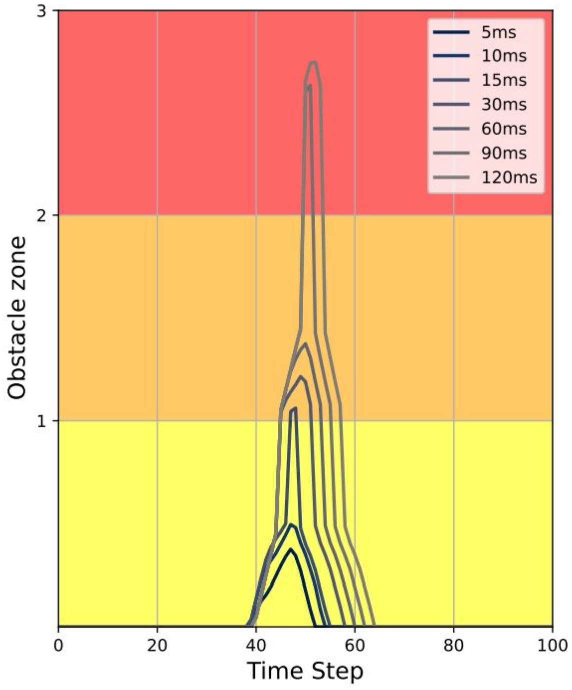
(b) UAV 1 path.

Remote API scenario

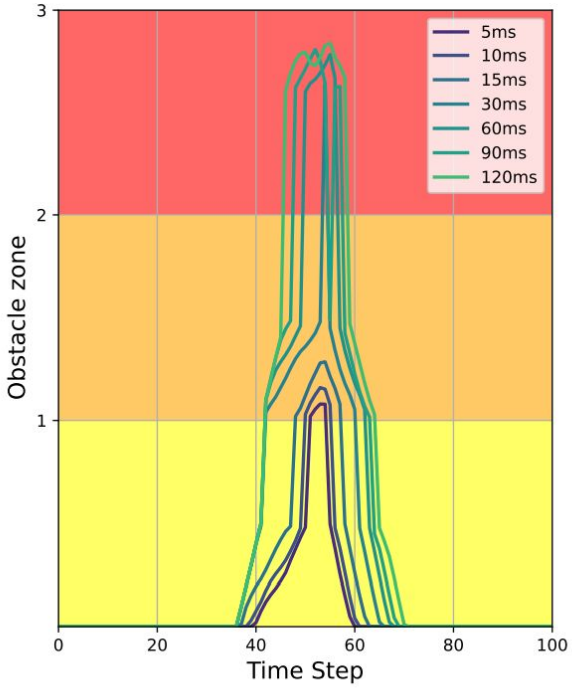




Remote API scenario

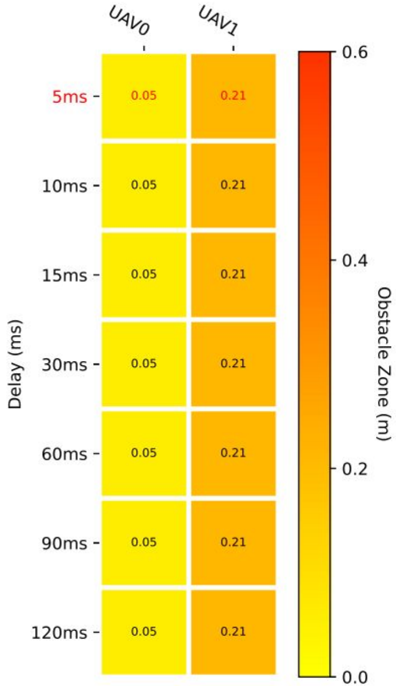


(a) UAV 0 path.

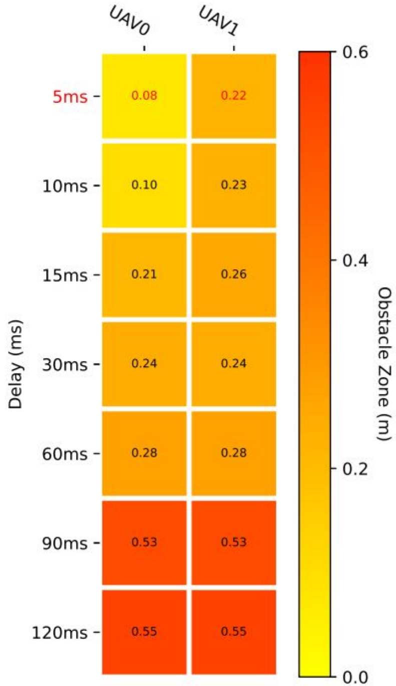


(b) UAV 1 path.

In-network control vs Remote API control

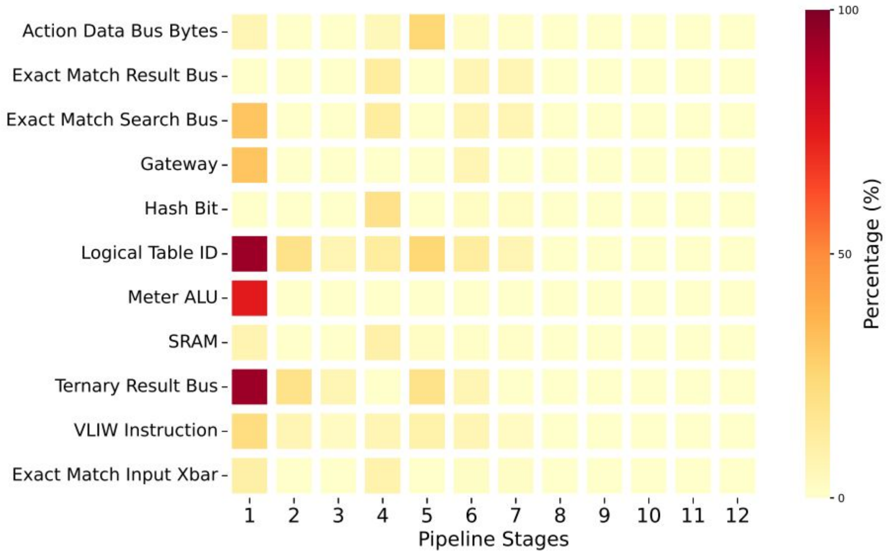


(a) In-network control.

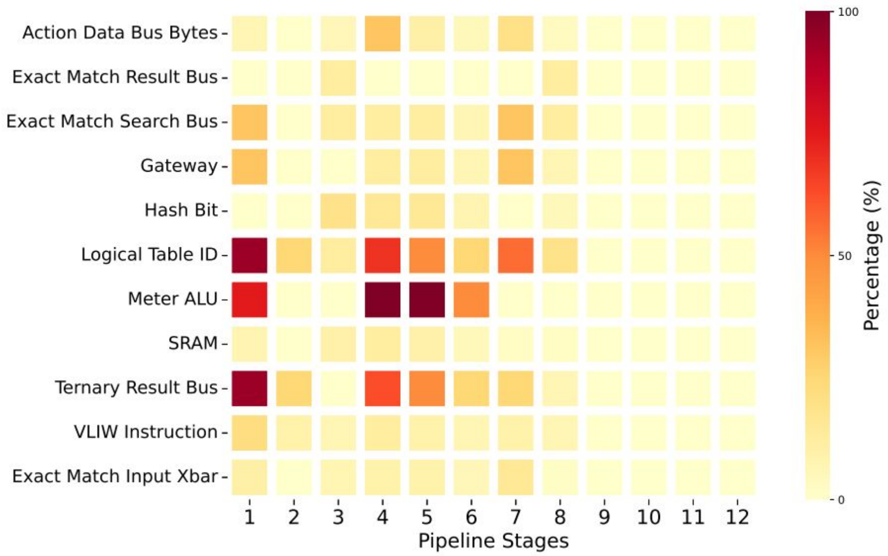


(b) Remote API control.

Resource Utilization



(a) Single UAV scenario (baseline)



(b) Five UAVs scenario

Resource Utilization

Resource	Switch.p4	P7.p4[a]	UAV.p4[b]	Fabric.p4
Hash Bits	0.9%	3.3%	2.9%	8.7%
SRAM	4.6%	4.5%	4.2%	12.4%
TCAM	8.3%	8.3%	9.3%	15.6%
VLIW Actions	1.0%	5.9%	5.7%	8.6%
Stateful ALUs	0.0%	0.0%	0.0%	29.2%
Gateway	0.5%	6.8%	3.6%	18.8%
Meter ALUs	0.0%	6.3%	6.3%	4.2%
Logical Table ID	1.0%	11.4%	15.6%	26.6%
Ternary Result Bus	0.5%	9.4%	12.5%	19.8%

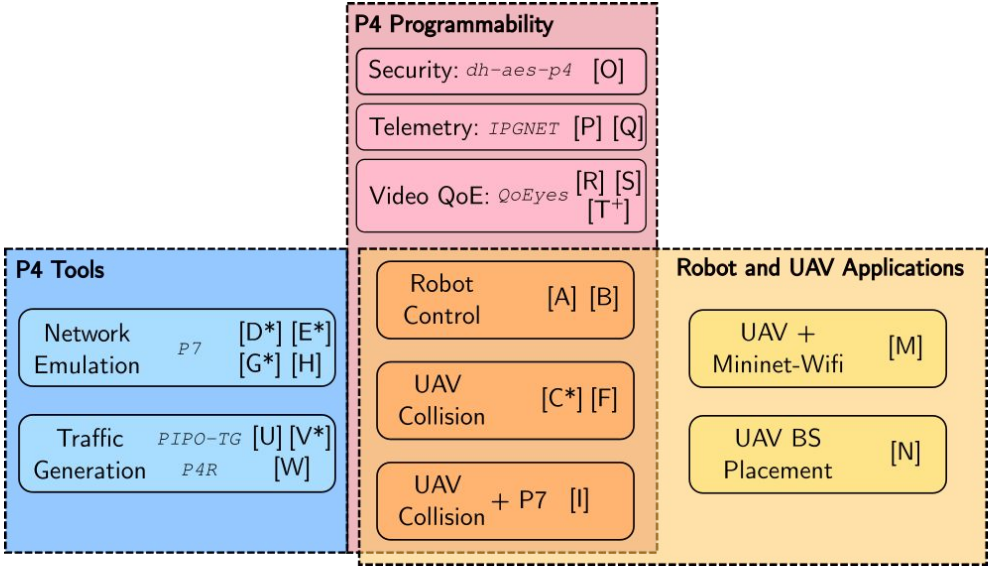
[a] One switch and two links topology
[b] Basic configuration of a single UAV

06.

Final Remarks

- Contributions & publications
- Conclusion & Future Work

Contributions & publications

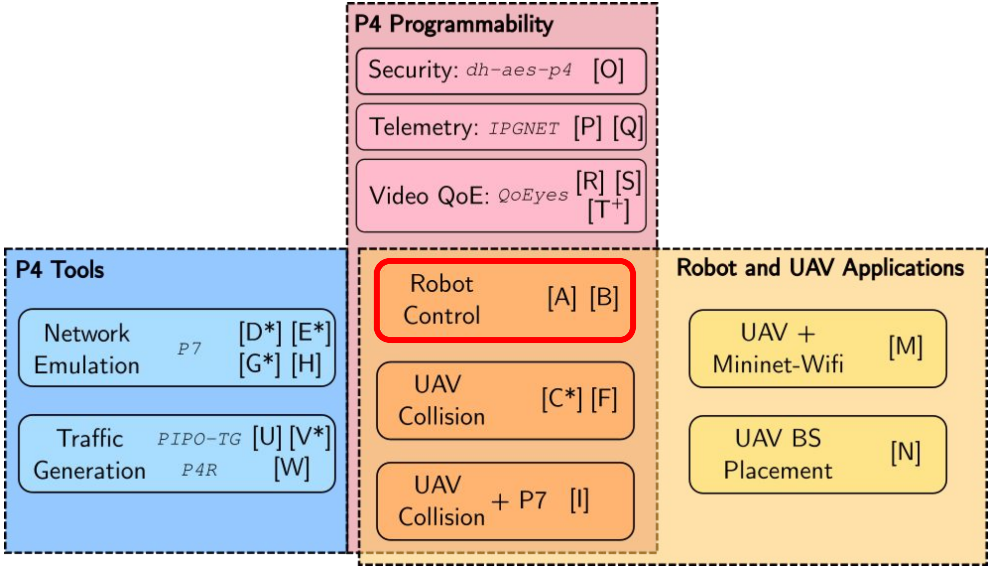


Contributions & publications

Main thesis contributions

A. F. Rodriguez, C. E. Rothenberg and G. Pongrácz. In-network p4-based low latency robot arm control. In: 15th CONEXT, Orlando, USA, December 2019.

B. F. Rodriguez, L. Csikor, C. Recalde, C. E. Rothenberg and G. Pongrácz. Towards low latency industrial robot control in programmable data plane. In: 2020 6th IEEE NetSoft, Virtual Conference, June 2020.

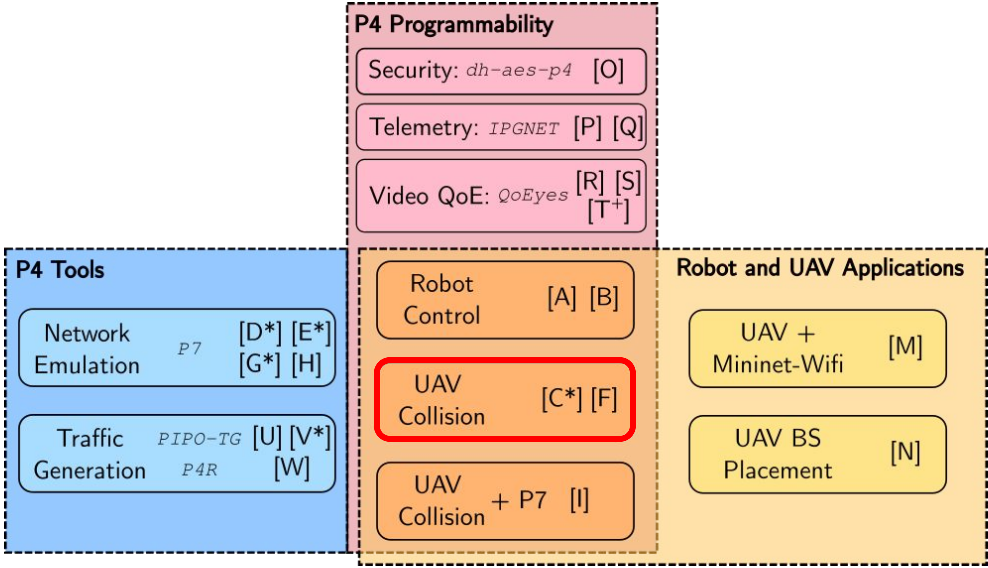


Contributions & publications

Main thesis contributions

C. F. Rodriguez, G. Szabó, C. E. Rothenberg and G. Pongrácz. Quadcopter Implementation of an in-network Centralized Collision Avoidance Algorithm in Programmable Data Planes. In: P4 Workshop 2021, Virtual Conference, May 2021. Most Novel Non-Networking Use Of P4 Award.

F. F. Rodriguez, G. Szabó, C. E. Rothenberg and G. Pongrácz. Towards an In-network UAV Centralized Collision Avoidance Algorithm in Programmable Data Planes. In: Globecom 2022 Demos, Rio de Janeiro, Brazil, December 2022.

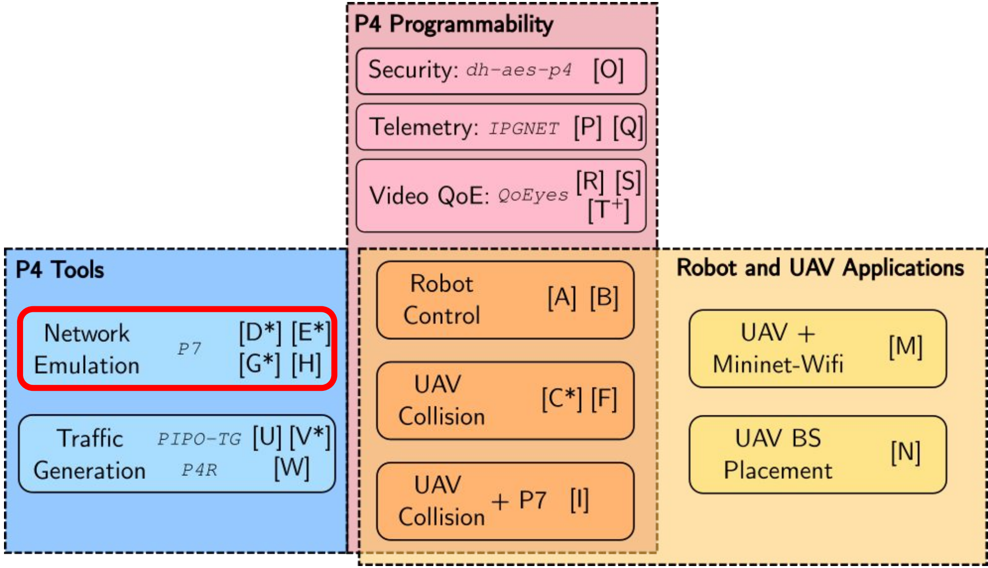


Contributions & publications

Main thesis contributions

D. F. Rodriguez, M. F. Schwarz and C. E. Rothenberg. P7 (P4 Programmable Patch Panel): An Instant 100g Emulated Network Testbed In A Pizza Box. In: P4 Workshop 2022, May 2022. Most Novel Application in a Networking Use Case Award.

E. F. Rodriguez, F. G. Vogt, A. G. Castro, M. F. Schwarz and C. E. Rothenberg. P4 Programmable Patch Panel (P7): An Instant 100G Emulated Network on Your Tofino-based Pizza Box. In: ACM SIGCOMM Posters, Demos, and SRC, Amsterdam, August 2022. SRC'22 Graduate Medal Award.

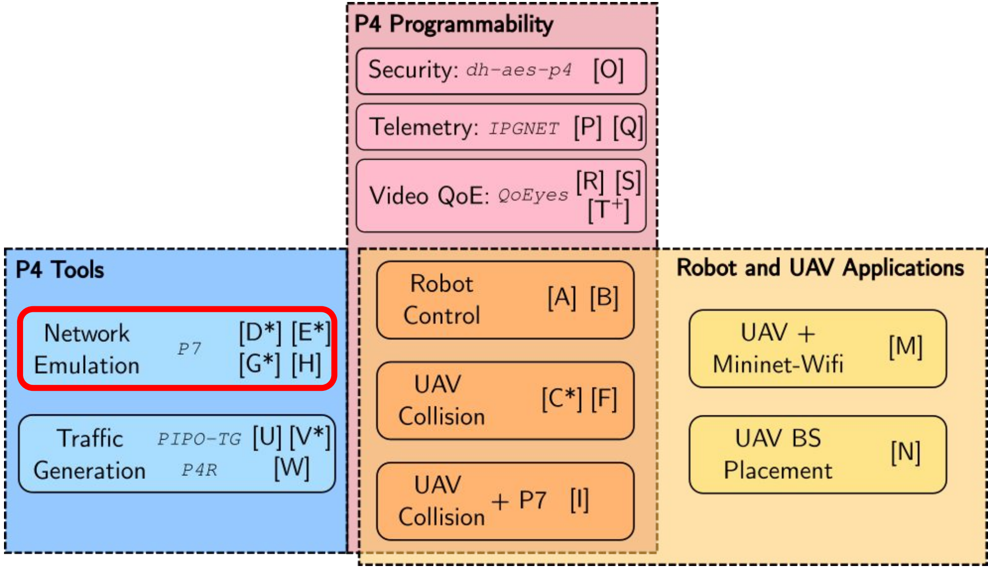


Contributions & publications

Main thesis contributions

G. F. Rodriguez, F. G. Vogt, A. G. Castro, M. F. Schwarz and C. E. Rothenberg. Network Emulation with P7: A P4 Programmable Patch Panel on Tofino-based Hardware. In: SBRC 2023 Salão de Ferramentas, Brasilia, Brazil, May 2023. Best demo Award.

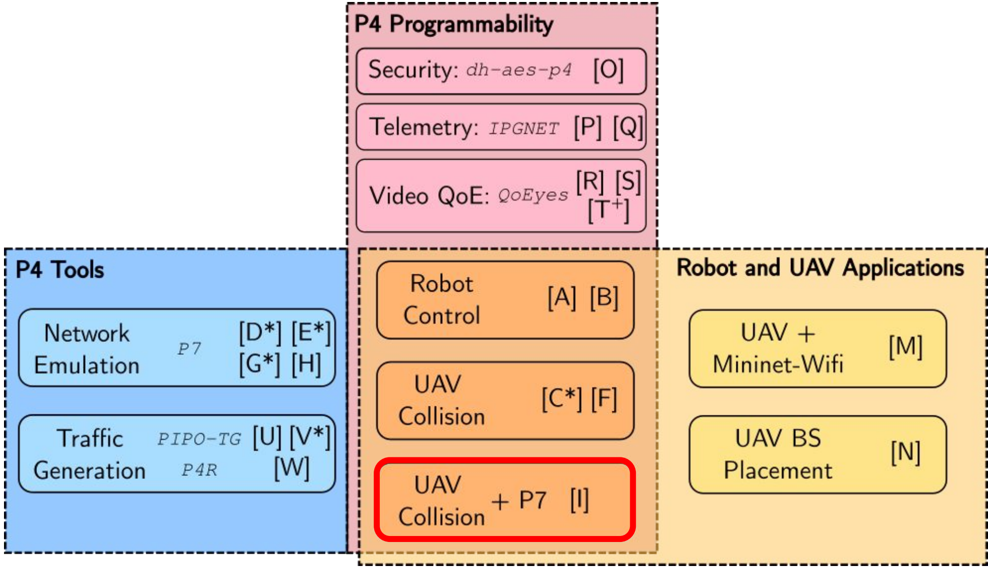
H. F. Rodriguez, F. G. Vogt, A. G. Castro, and C. E. Rothenberg. Towards Multiple Pipelines Network Emulation with P7. In: NetSoft 2023 Demos, Madrid, Spain, June 2023.



Contributions & publications

Main thesis contributions

I. F. Rodriguez, and C. E. Rothenberg. Offloading Robotic and UAV Applications to the Network Using Programmable Data Planes. In: NFV-SDN Doctoral Symposium, Dresden, Germany, November 2023.

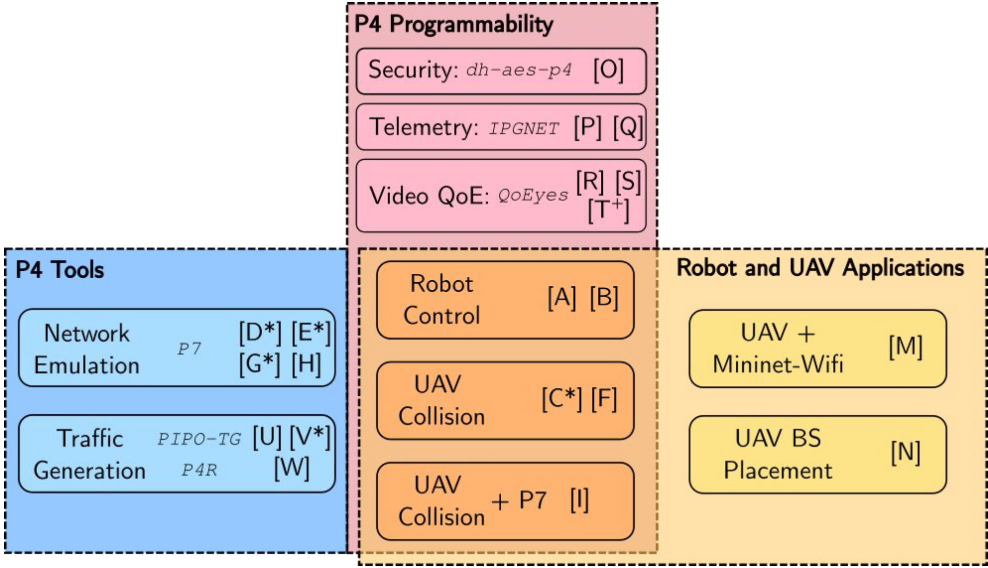


Contributions & publications

Scientific results based on our work

J. Leonardo Rodrigues Marques. Realistic packet loss model for a programmable dataplane testbed. Final Course Work (TFC) (Electrical Engineering) - University of Campinas, Dec 2022.

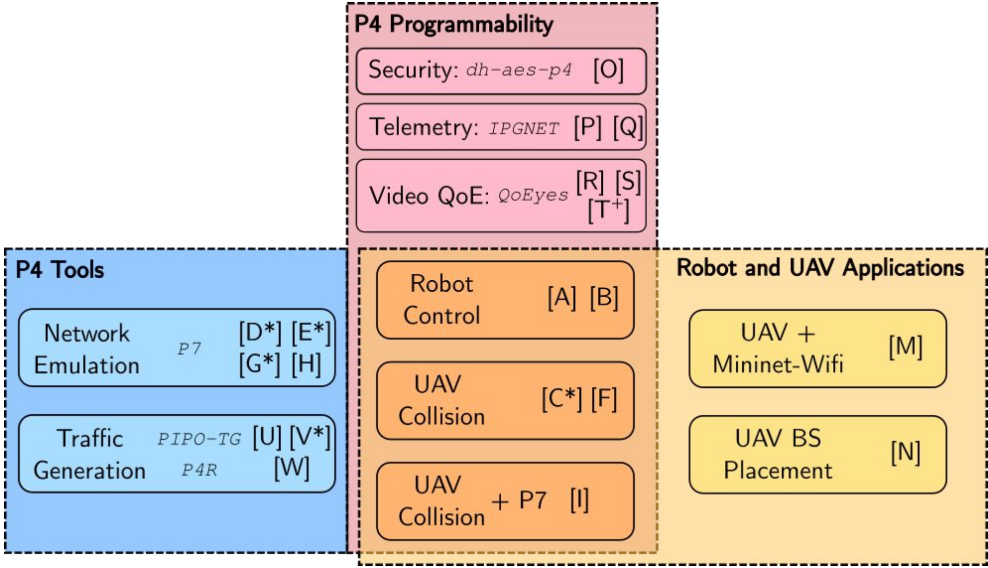
K. Carlos Rony Recalde Vitonera. Edge Data Plane Programmability for Robotic Control Application Latency Improvements. MSc Dissertation - University of Campinas. Advisor: Christian Esteve Rothenberg. (13/06/2023)



Contributions & publications

Scientific results based on our work

L. Francisco Germano Vogt. In-band telemetry using inter-packet gap metrics for network and service management. MSc Dissertation - University of Campinas. Advisor: Christian Esteve Rothenberg, Jun 2023.

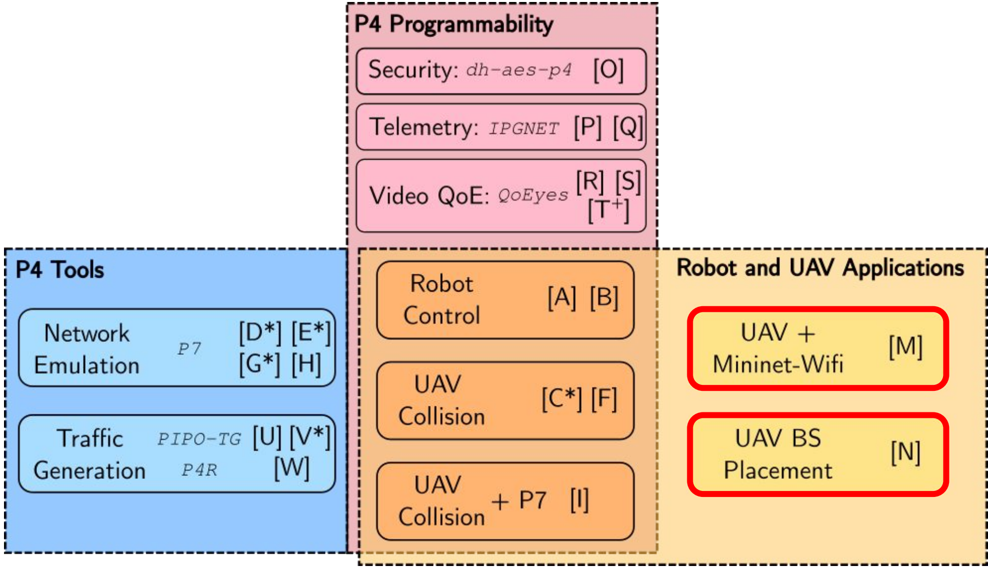


Contributions & publications

UAV Applications

M. F. Rodriguez and R. Fontes. Example of Drones integration using CoppeliaSim. In: Mininet-WiFi, GitHub, January 2021.

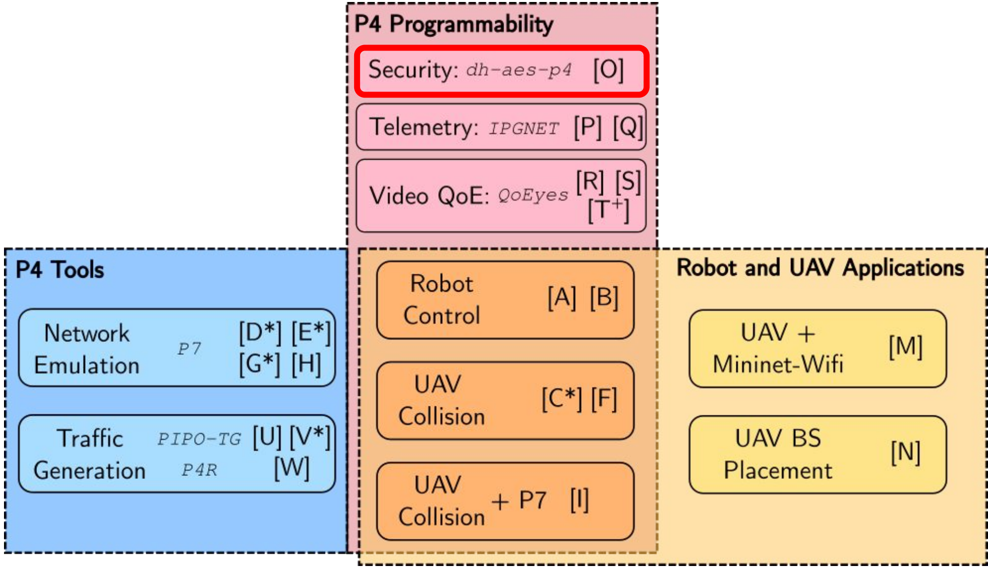
N. F. Pasandideh, F. Rodriguez, P. H. M. Pereira, C. E. Rothenberg and E. P. Freitas. An improved Particle Swarm Optimization algorithm for UAV Base Station placement. In: Wireless Personal Communications Journal, March 2023.



Contributions & publications

P4 use cases

O. I. Oliveira, E. Neto, R. Immich, R. Fontes, A. Neto, Augusto and F. Rodriguez and C. E. Rothenberg. dh-aes-p4: On-premise encryption and in-band key-exchange in P4 fully programmable data planes. In: 2021 IEEE NFV-SDN, Virtual Conference, Nov 2021.

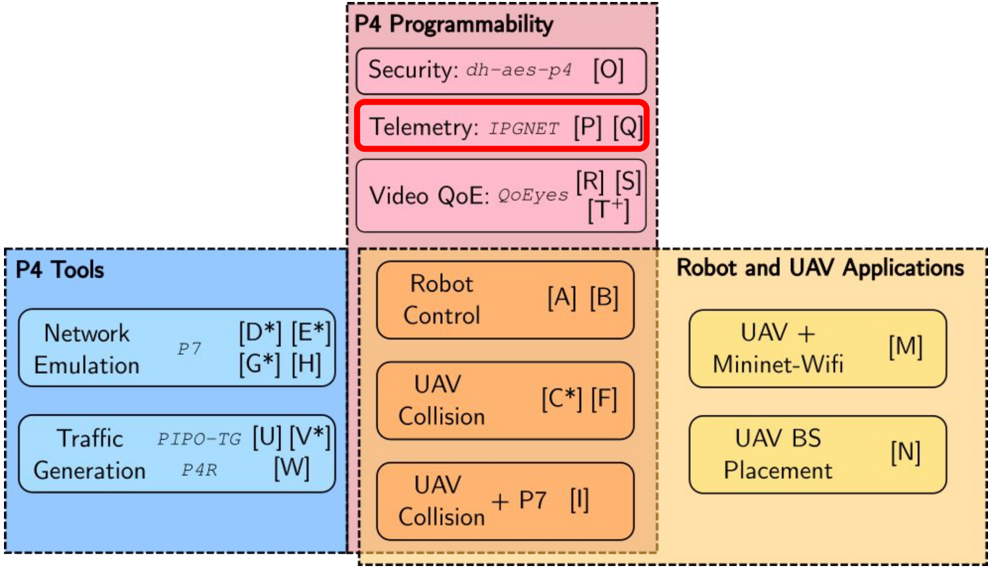


Contributions & publications

P4 use cases

P. F. G. Vogt, F. Rodriguez, C. E. Rothenberg and G. Pongrácz. In-band Inter Packet Gap Telemetry (IPGNET): Unlocking novel network monitoring methods. In: Globecom 2022 Demos, Rio de Janeiro, Brazil, December 2022.

Q. F. G. Vogt, F. Rodriguez, C. E. Rothenberg and G. Pongrácz. Innovative network monitoring techniques through In-band Inter Packet Gap Telemetry (IPGNET). In: Conext 2022 Euro P4 2022, Rome, Italy, December 2022.

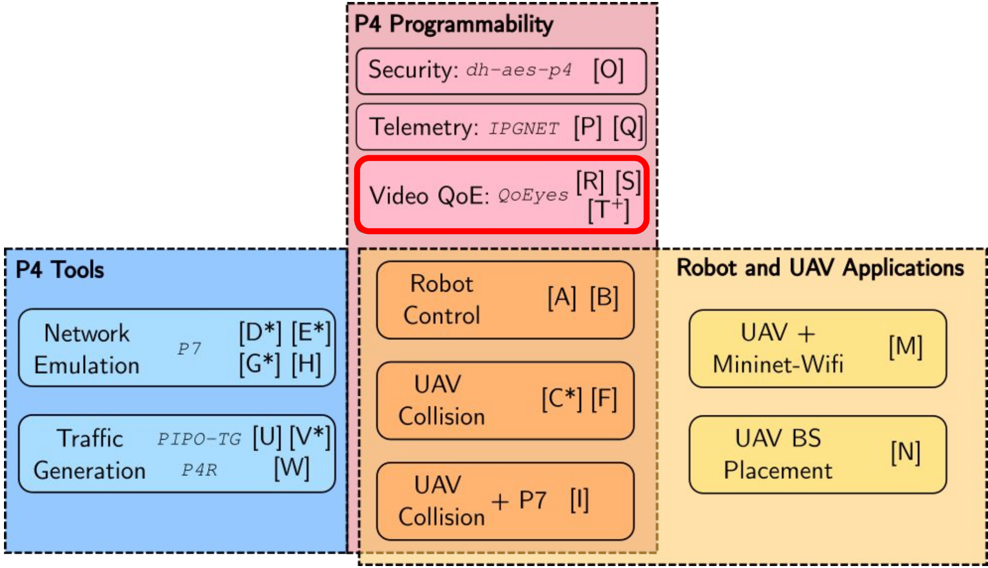


Contributions & publications

P4 use cases

R. F. G. Vogt, F. Rodriguez, C. E. Rothenberg and G. Pongrácz. QoEyes: Towards Virtual Reality Streaming QoE Estimation Entirely in the Data Plane. In: NetSoft 2023, Madrid, Spain, June 2023.

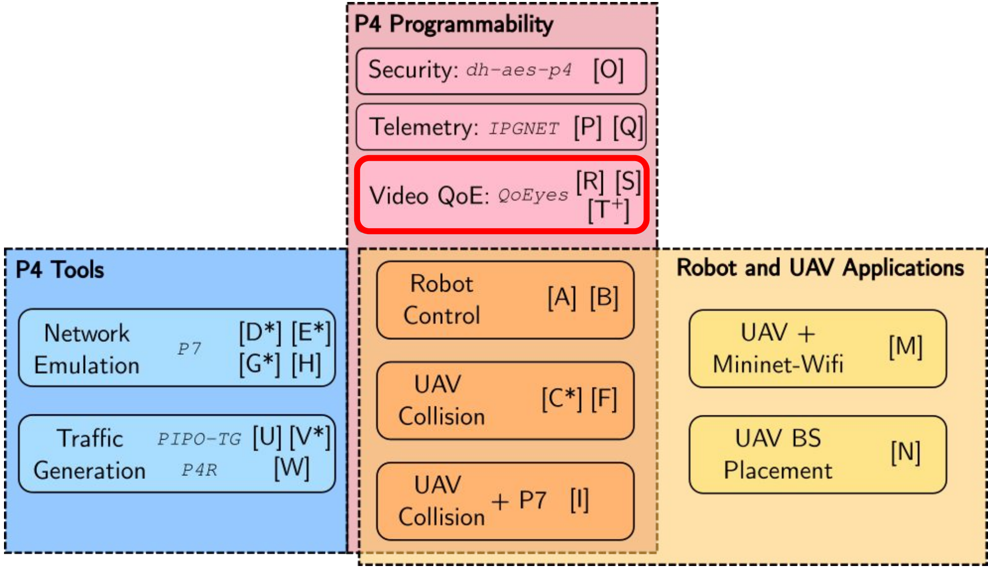
S. F. G. Vogt, F. Rodriguez, C. E. Rothenberg and G. Pongrácz. Demo of QoEyes: Towards Virtual Reality Streaming QoE Estimation Entirely in the Data Plane. In: NetSoft Demos, Madrid, Spain, June 2023.



Contributions & publications

P4 use cases

T. F. G. Vogt, F. Rodriguez, A. G. Castro, S. K. Singh, M. C. Luizelli, C. E. Rothenberg and G. Antichi. Video Streaming QoE Meets Programmable Data Planes: In-Network QoE Management. In: IEEE Network Magazine Advancements in Network-Assisted Video Streaming: Optimization and Performance Analysis.

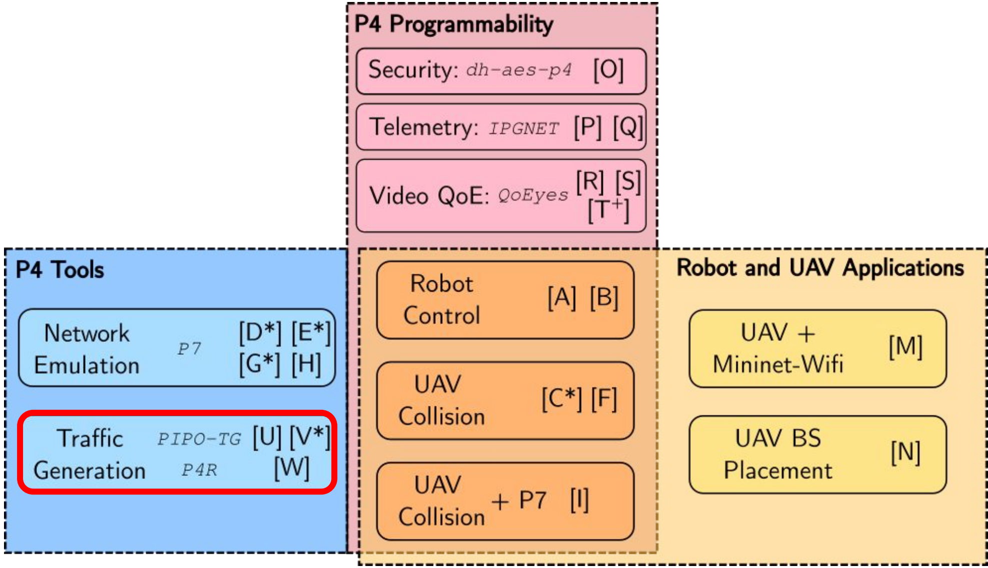


Contributions & publications

P4 use cases

U. F. G. Costa, F. G. Vogt, F. Rodriguez, A. G. Castro, M. C. Luizelli, and C. E. Rothenberg. PIPO-TG: Parameterizable High-Performance Traffic Generation. In: NOMS 2024, Seoul, South Korea, May 2024.

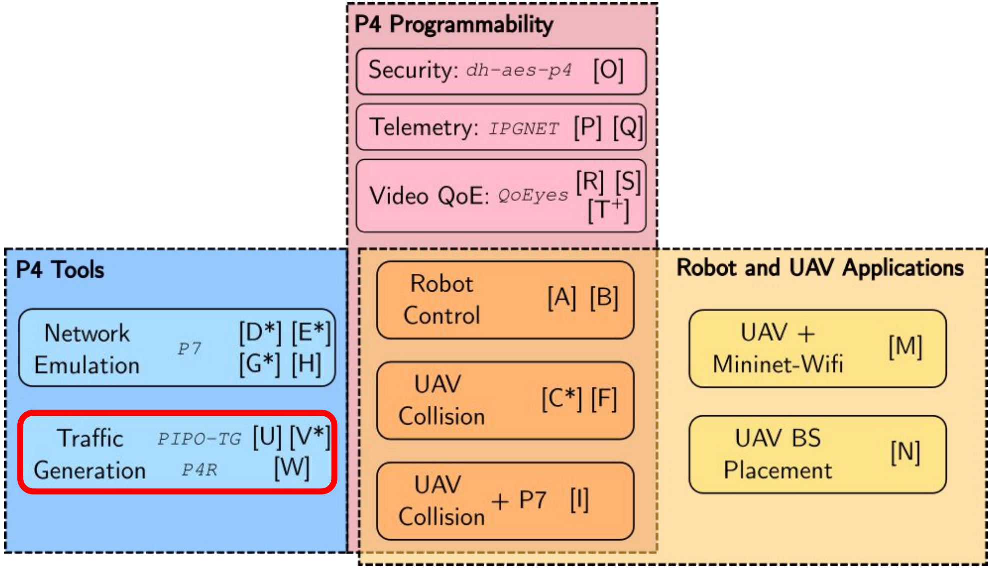
V. F. G. Costa, F. G. Vogt, F. Rodriguez, M. C. Luizelli, and C. E. Rothenberg. Towards Time-Sensitive Networking Traffic Generation with PIPO-TG. In: SBRC 2024, Niterói, Brazil, May 2024. Best demo award runner-up.



Contributions & publications

P4 use cases

Q. F. G. Vogt, F. Rodriguez, F. G. Costa, M. C. Luizelli, C. E. Rothenberg, G. Patra, and G. Pongrácz. Demo: P4 Replay (P4R): Reproducing Packet Traces and Stateful Connections at Line-Rate on Your P4-capable Hardware. In: SIGCOMM 2024 Demos, Australia, August 2024.



Contributions & publications

Available open-source repositories

Main thesis contributions

- https://github.com/intrig-unicamp/p4_robot
- https://github.com/intrig-unicamp/p7
- https://github.com/intrig-unicamp/uav

UAV Applications

- https://github.com/intrig-unicamp/mininet-wifi/tree/master/examples/uav

P4 use cases

- https://github.com/intrig-unicamp/QoEyes
- https://github.com/intrig-unicamp/IPGNET
- https://github.com/intrig-unicamp/PIPO-TG
- https://github.com/intrig-unicamp/P4R

Final Considerations

- We offload some controller functions to a programmable edge network device.
- We manipulate the content of the messages to craft new replies within an established TCP session.
- We validate the impact of the delay in the connection.

Open Opportunities

- Multiple robot synchronization.
- Applications to a wide range of applications (e.g., industry 4.0, automation, real-time control, synchronization, drones).

Final Considerations

- P7 supports multiple pipelines, contributing to experimental platforms that support data plane programmability.
- We present a solution that allows the user to create a traditional networking topology and advanced programmable networking scenarios.
- We offer a user-friendly environment, a cost-effective 100G network emulator, and a hardware-based solution for research and teaching.

Open Opportunities

- Address open challenges (i.e., Buffers consumption)
- New features (i.e., In-band Network Telemetry (INT), Dynamic link behaviors, Trace base link characteristics).
- Embed into disaggregated network testbed initiatives.

Final Considerations

- We propose an in-network collision avoidance algorithm in edge Programmable Data Planes.
- We explore the benefits of a P4 device (e.g., High Performance, Reconfigurability, Protocol Independent) in a ULL use case.
- The collision avoidance algorithm in P4 can detect and react fast to avoid possible impacts effectively.

Open Opportunities

- Enhance the environment, including other variables (e.g., 5G RAN, UPF).
- Add new in-network functions in the programmable data plane (e.g., Path planning algorithm).

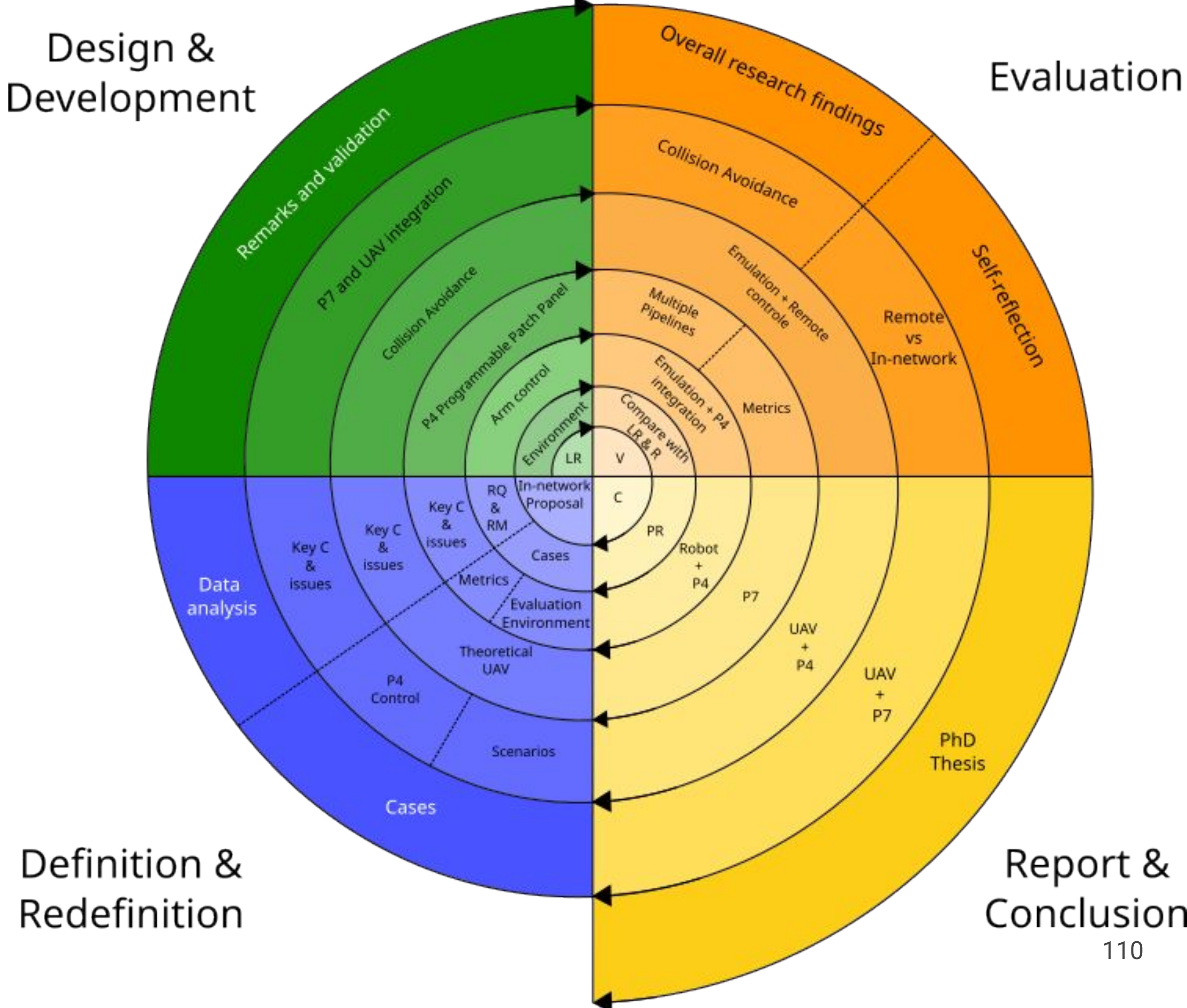
Thank You

PhD candidate: Fabricio Rodriguez
Advisor: Prof. Christian Esteve Rothenberg

07/10/2024

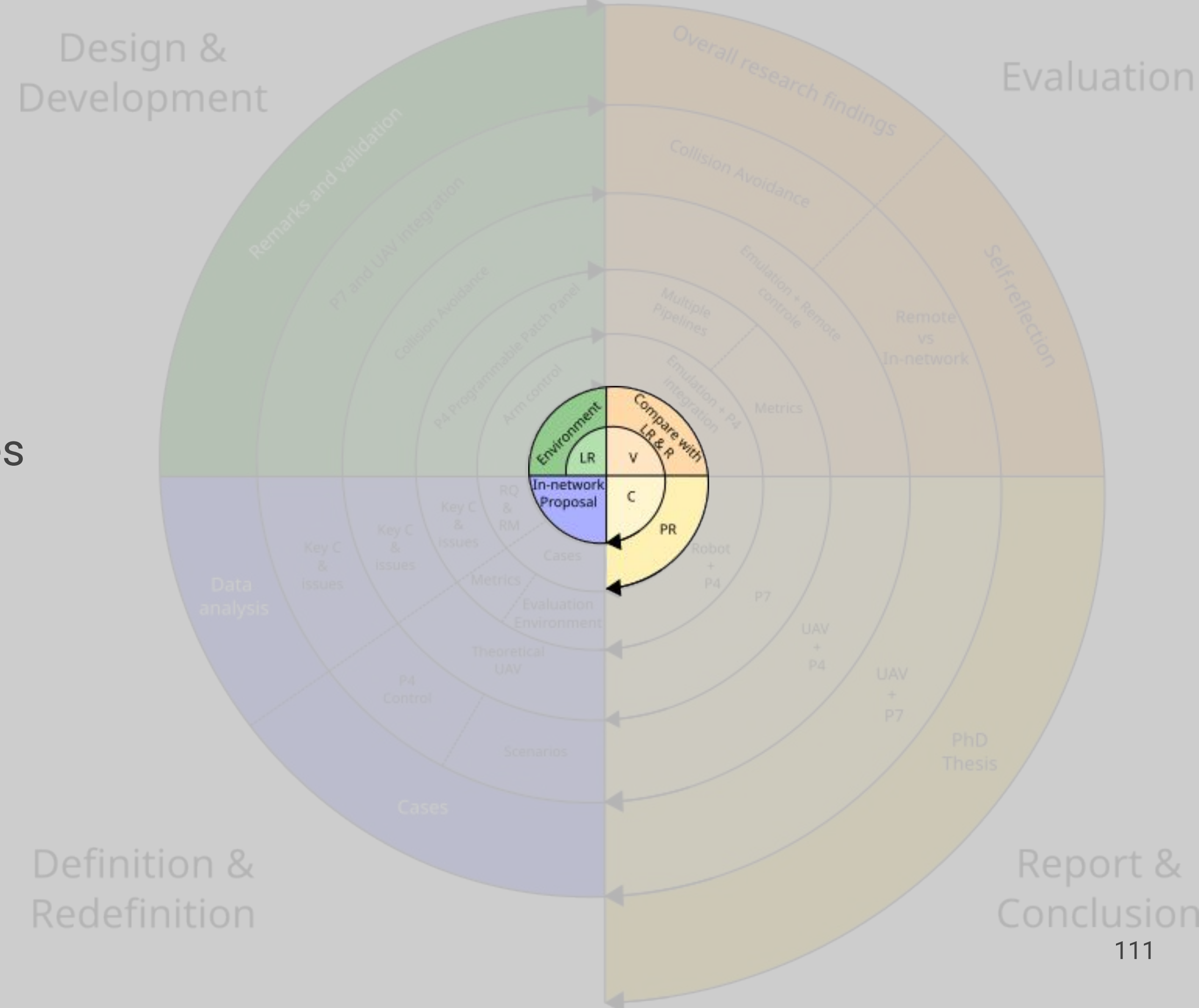
INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

Spiral model methodology



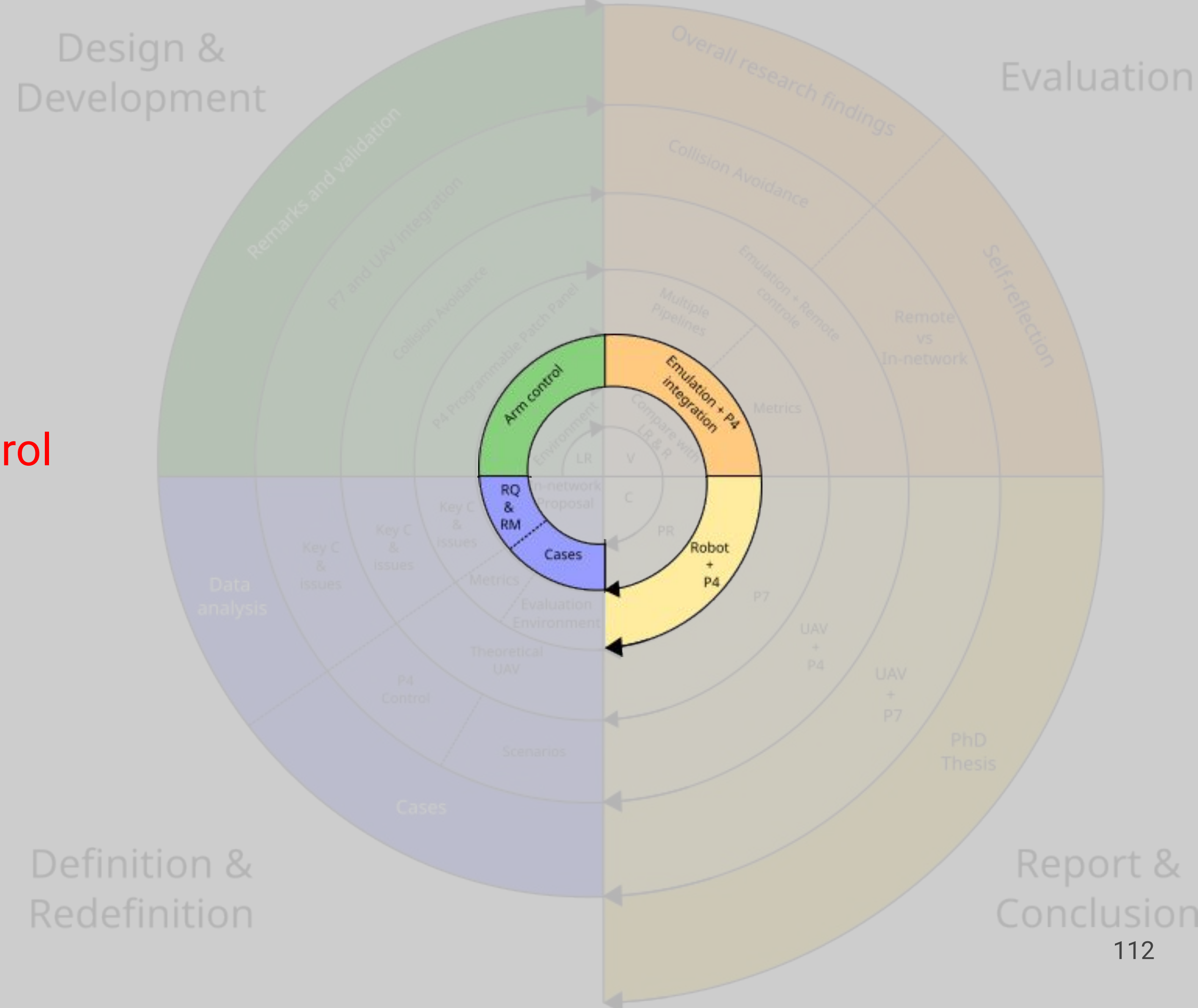
Spiral model methodology

- Literature Review
- In-network use cases
- Preliminary Results



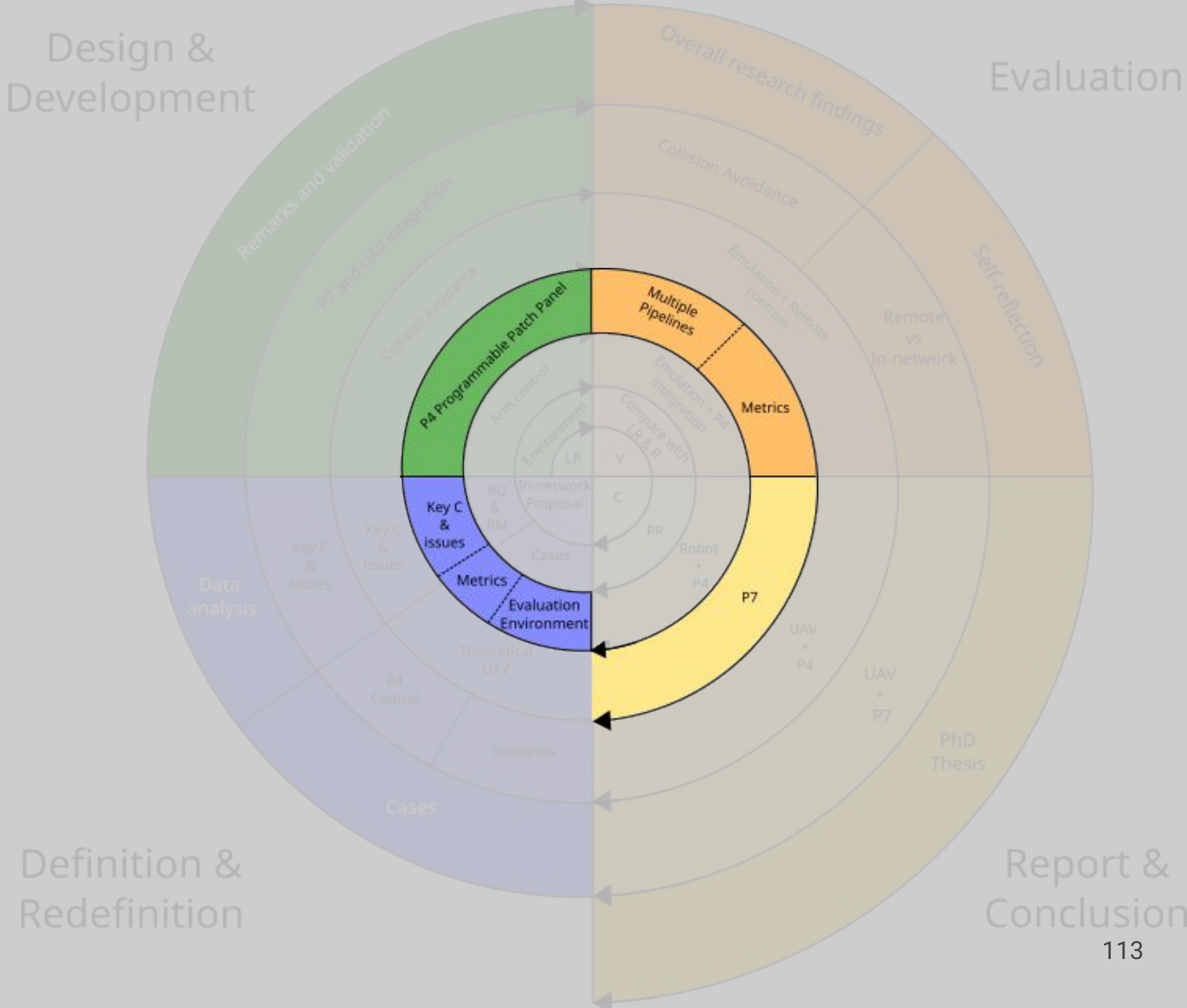
Spiral model methodology

- Arm control
- In-network arm control



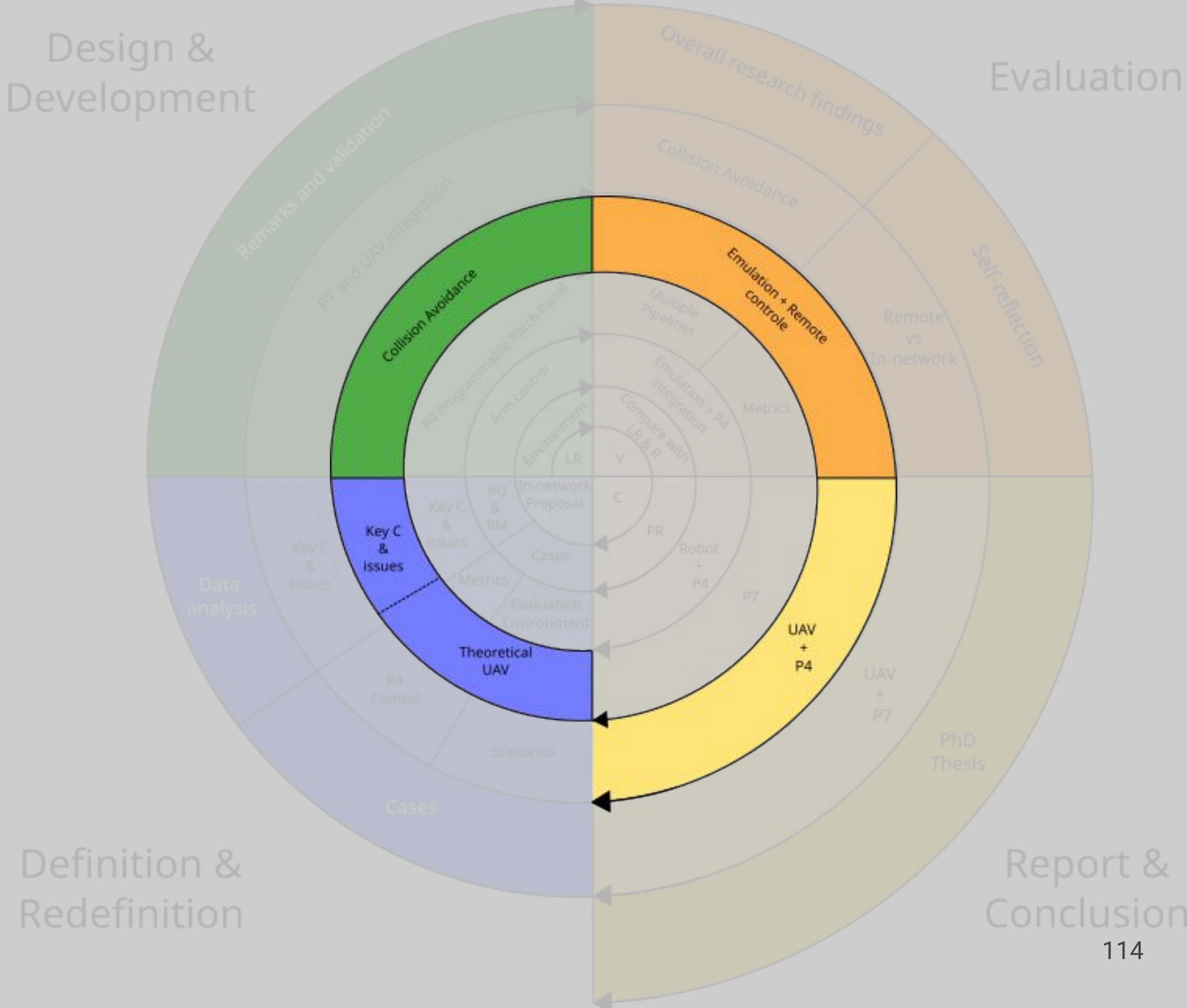
Spiral model methodology

- Evaluation environment
- Network metrics
- P4 Programmable Patch Panel (P7)



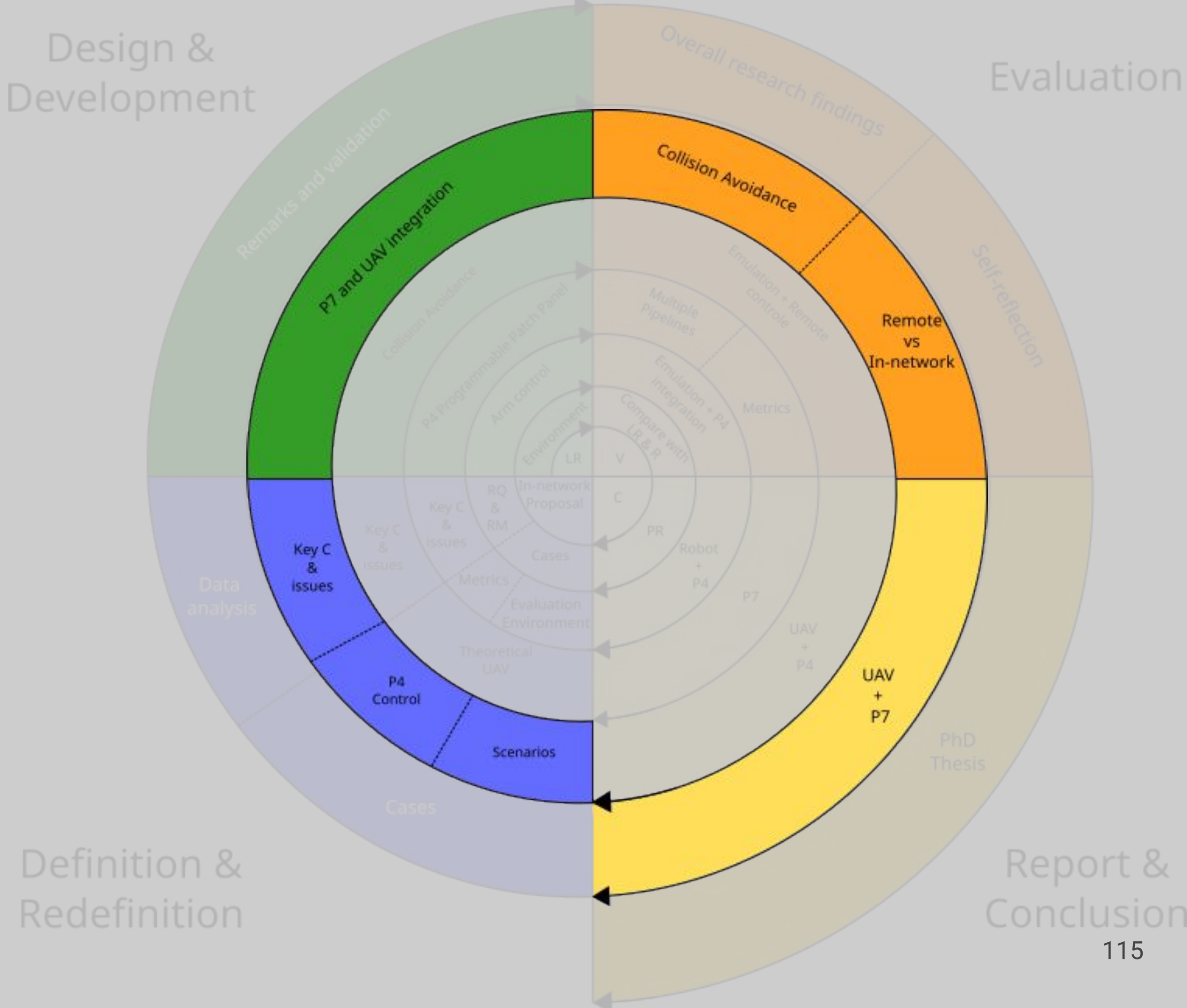
Spiral model methodology

- Collision Avoidance
- In-network UAV collision avoidance algorithm



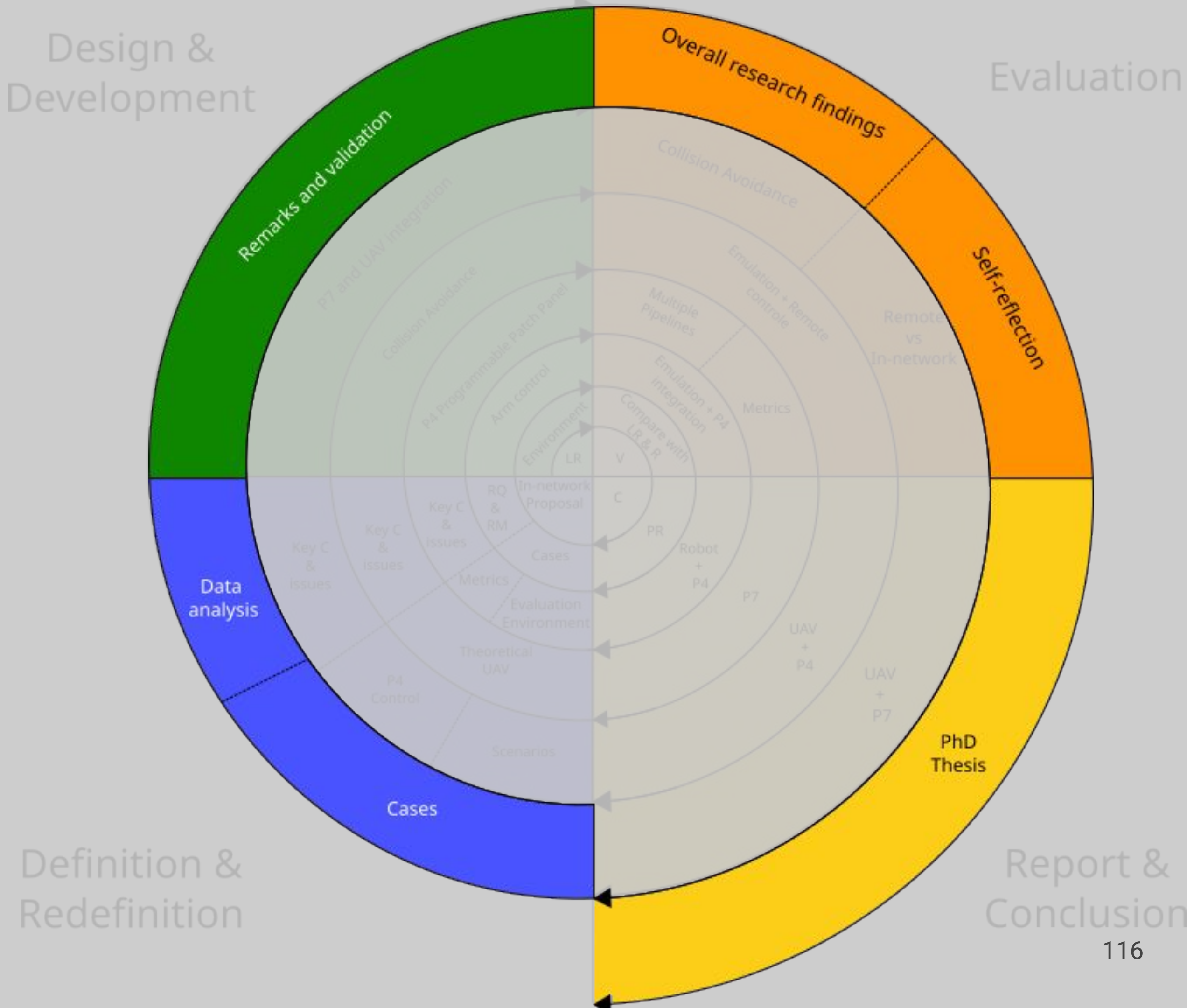
Spiral model methodology

- UAV scenarios
- Integration
- Collision avoidance algorithm over P7



Spiral model methodology

- Final use cases
- Research findings
- PhD thesis



Final Considerations

- In-network function = rapidly detect and react
  - Controller functions in edge network device.
  - Impact of the latency in end-to-end connection.
  - User-server communication.
  - Deploy collaborative robots alongside human workers.
  - Collision avoidance algorithm in P4 can avoid potential impacts effectively.
  - P4 = low latency.

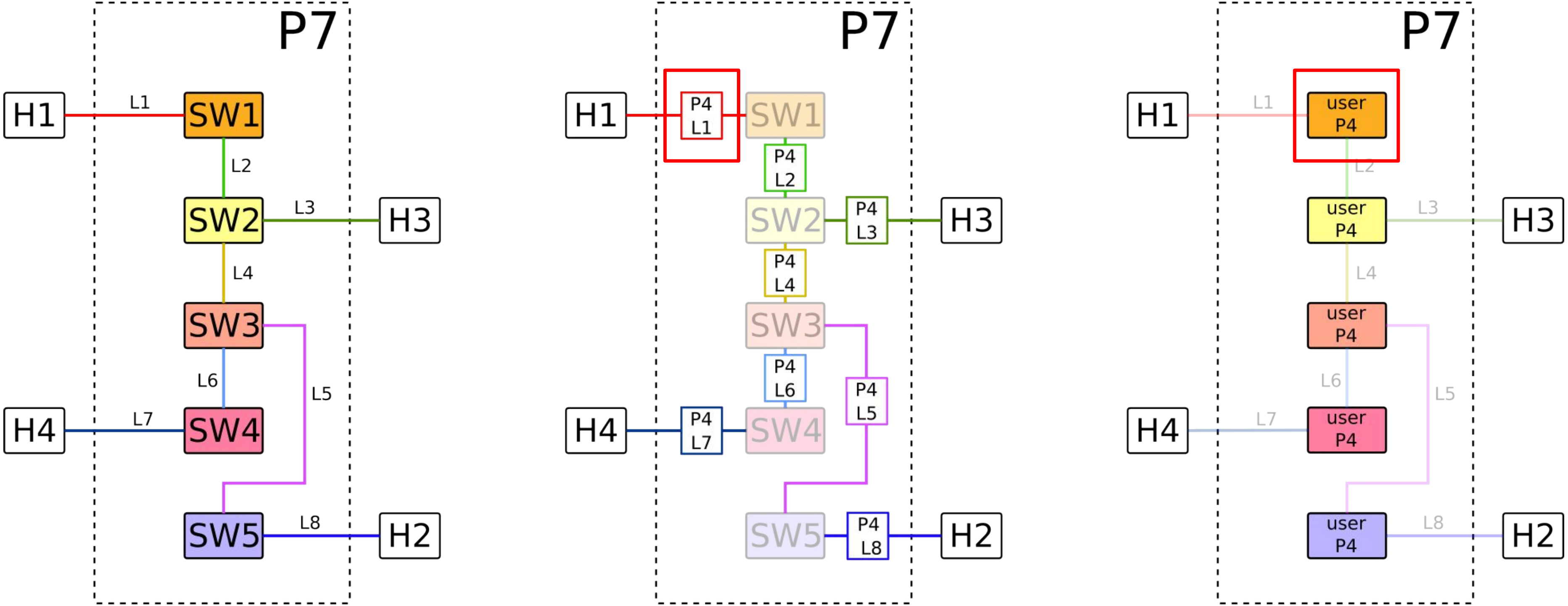
- Network emulation
  - Hardware-based solution.
  - Traditional networking topologies.
  - Advanced programmable networking scenarios.
  - Multiple pipelines.
  - 100G network emulator.

Final Considerations

Open Opportunities

- Increasing collaborative interactions.
- New applications such as food service and healthcare.
- New algorithms such as Path planning.
- + Control plane.
- SmartNics + ML .
- Digital twin + network environment.
- Industrial networks emulation.
- In-band Network Telemetry (INT).

P7 topology

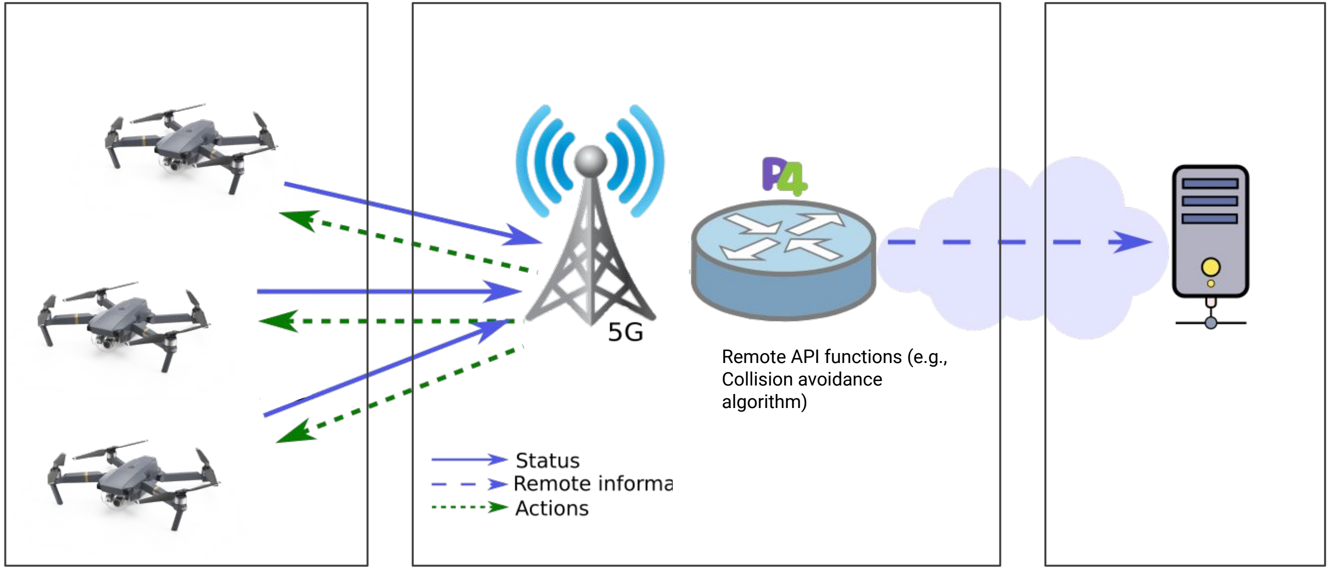

P7
H1
L1
SW1
L2
SW2
L3
H3
L4
SW3
L6
L5
H4
L7
SW4
SW5
L8
H2
P7
H1
P4 L1
SW1
P4 L2
SW2
P4 L3
H3
P4 L4
SW3
P4 L6
P4 L5
H4
P4 L7
SW4
SW5
P4 L8
H2
P7
H1
L1
user P4
L2
user P4
L3
H3
L4
user P4
L6
L5
H4
L7
user P4
user P4
L8
H2

Testbed

UAV

Edge P4 router

Remote API

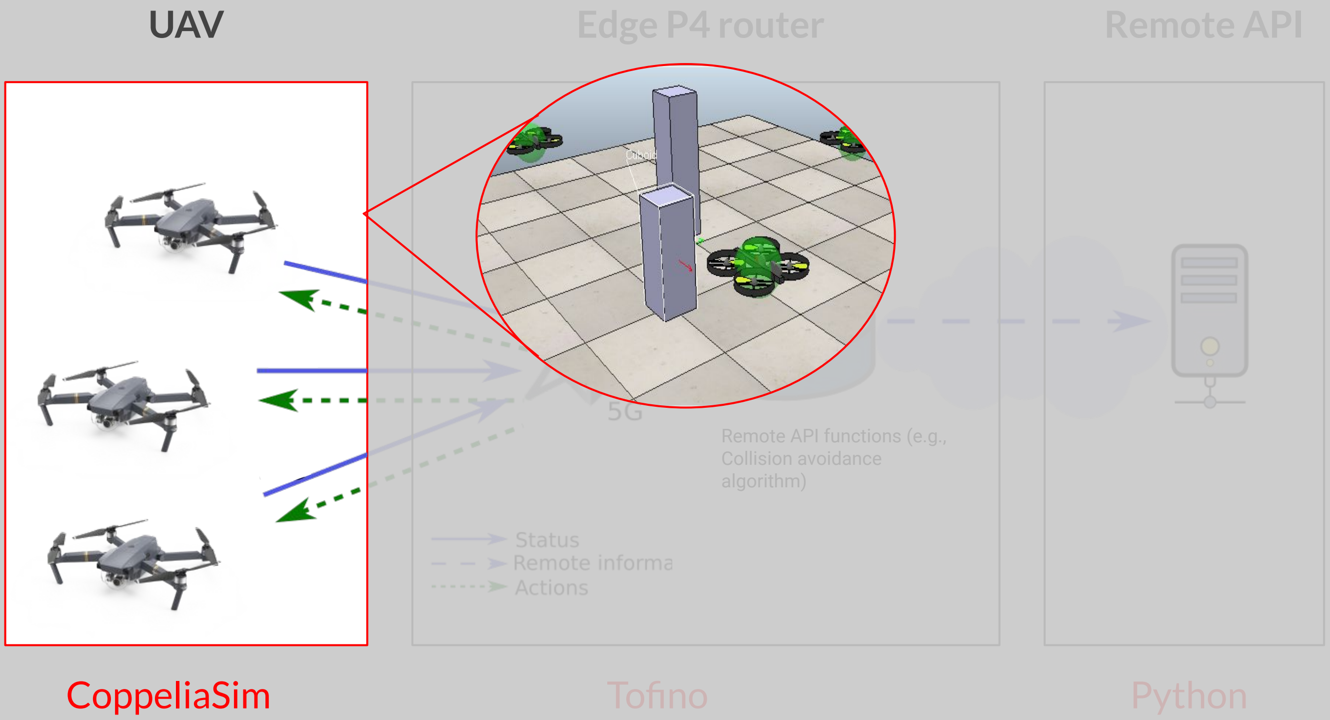


CoppeliaSim

Tofino

Python

Testbed

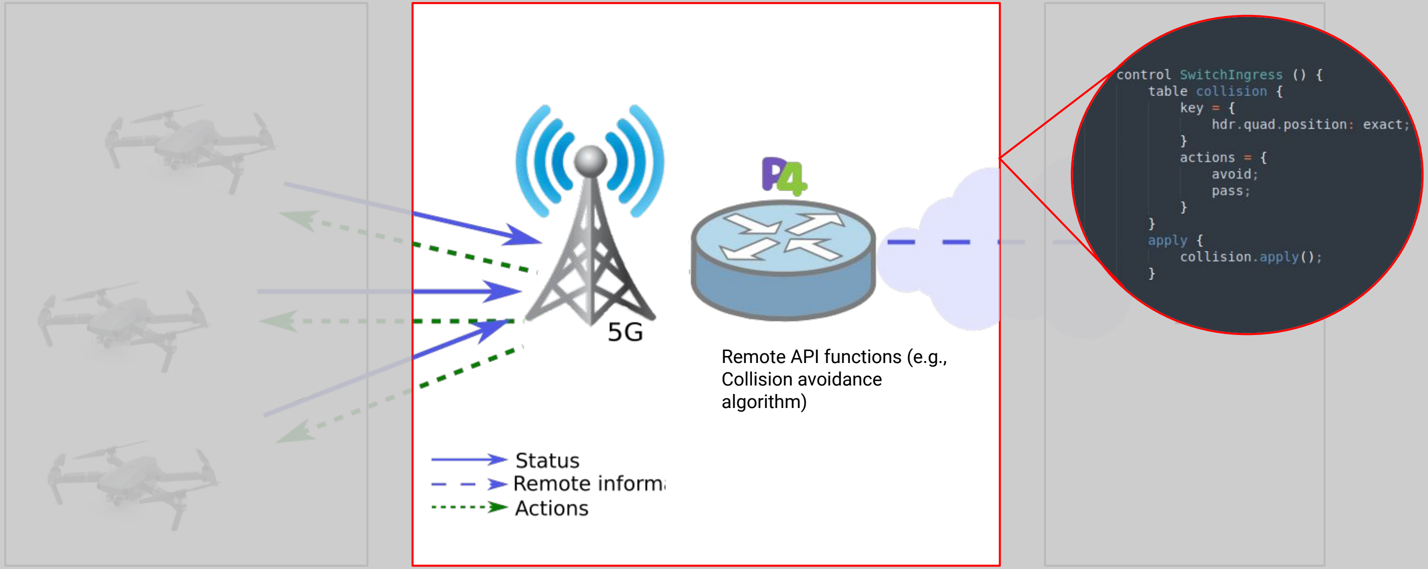

UAV
Edge P4 router
Remote API
5G
Remote API functions (e.g., Collision avoidance algorithm)
Status
Remote informa
Actions
CoppeliaSim
Tofino
Python

Testbed

UAV

Edge P4 router

Remote API

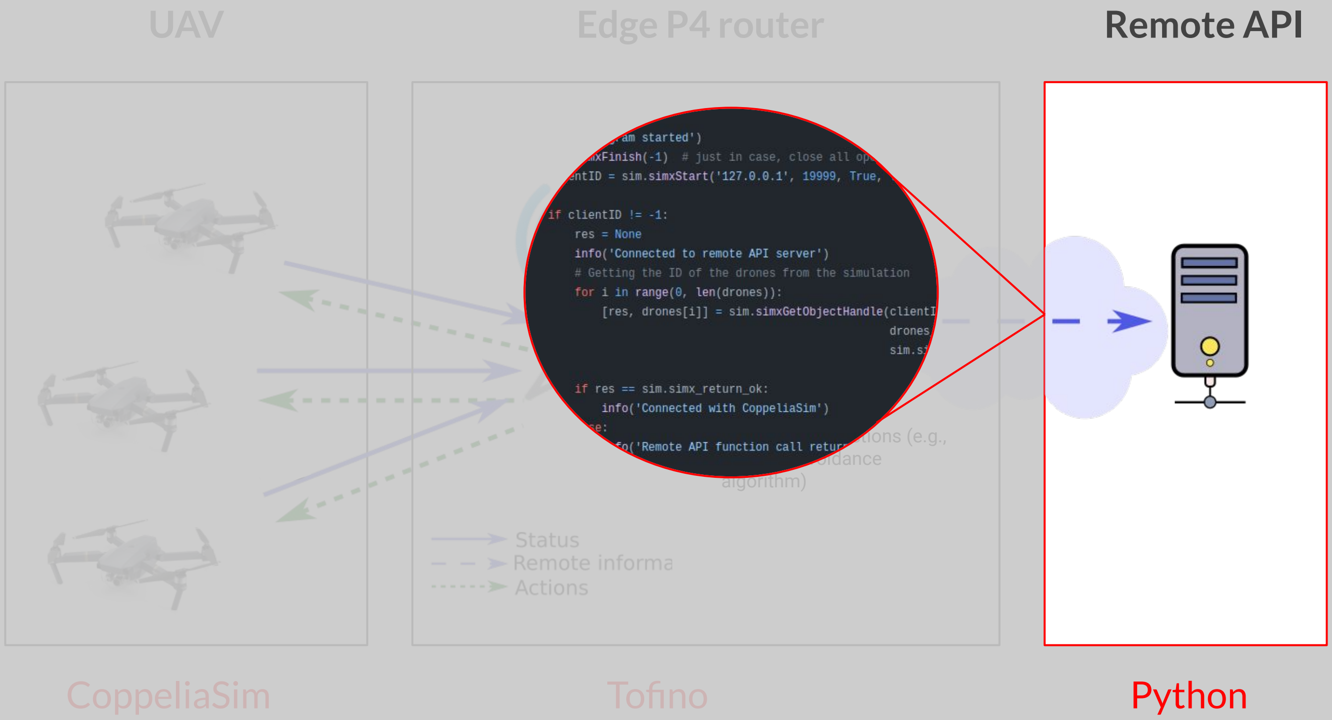


CoppeliaSim

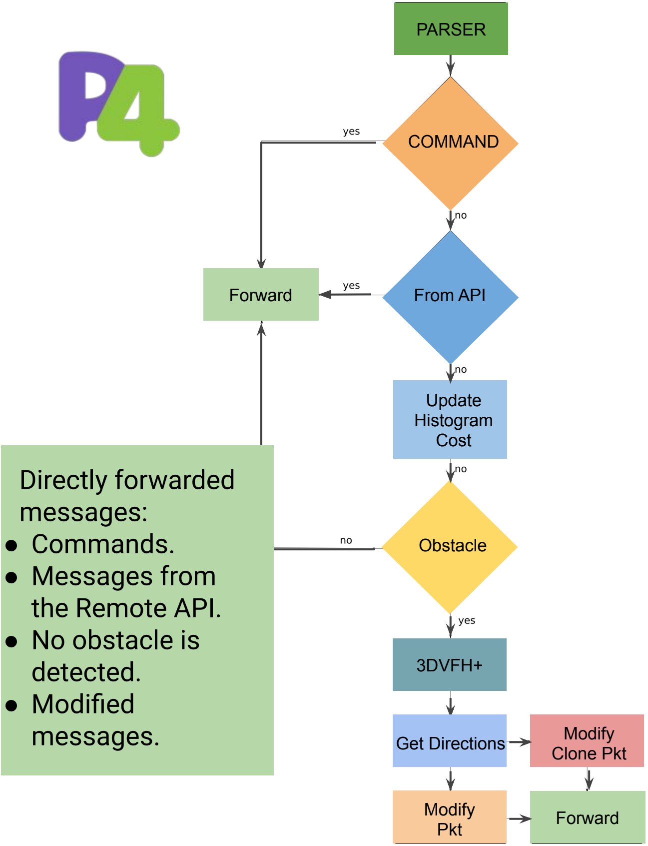
Tofino

Python

Testbed


UAV
Edge P4 router
Remote API
am started')
mxFinish(-1) # just in case, close all op
entID = sim.simxStart('127.0.0.1', 19999, True,
if clientID != -1:
res = None
info('Connected to remote API server')
# Getting the ID of the drones from the simulation
for i in range(0, len(drones)):
[res, drones[i]] = sim.simxGetObjectHandle(clientI
drones
sim.si
if res == sim.simx_return_ok:
info('Connected with CoppeliaSim')
fo('Remote API function call retur
Status
Remote informa
Actions
CoppeliaSim
Tofino
Python

3DVFH+ avoidance algorithm in P4



Extract the packet headers of the messages.

It is verified if the message contains a Command.

It is verified if the message comes from the Remote API.

The Histogram is Updated with the information of the packet.

With the updated information. It is verified if there is an obstacle nearby.

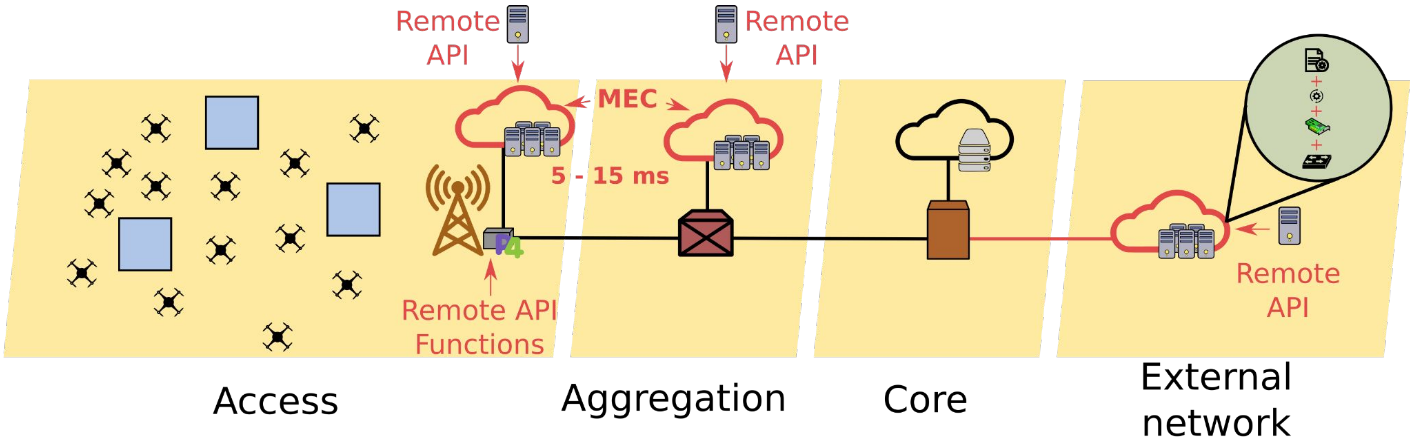
If an obstacle is detected, the 3DVFH+ algorithm starts.

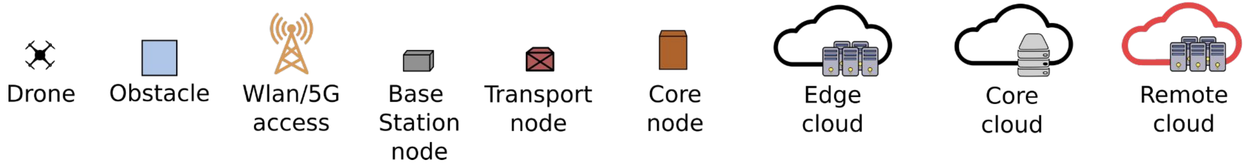
A new position is calculated (original position + safe range) to avoid the collision.

The original packet is updated.

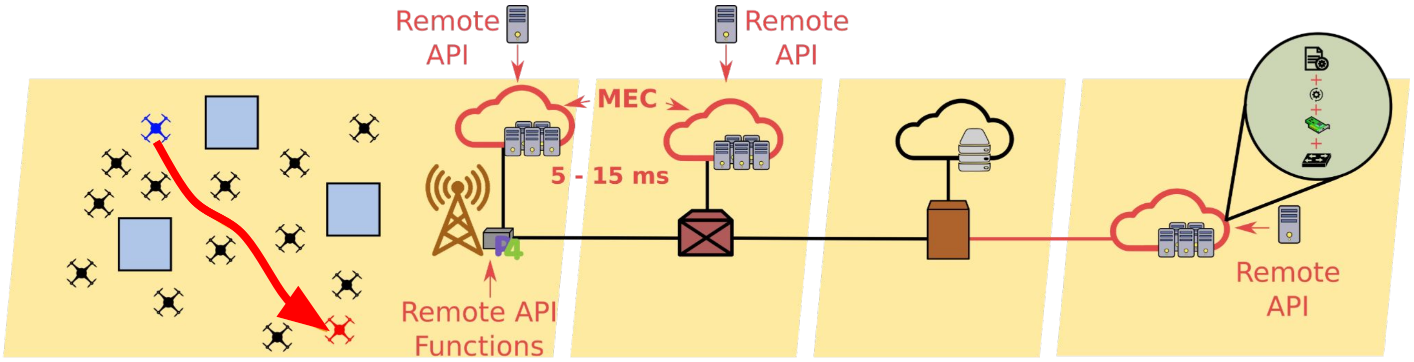
A clone packet is modified to be used as a command to set the new position.

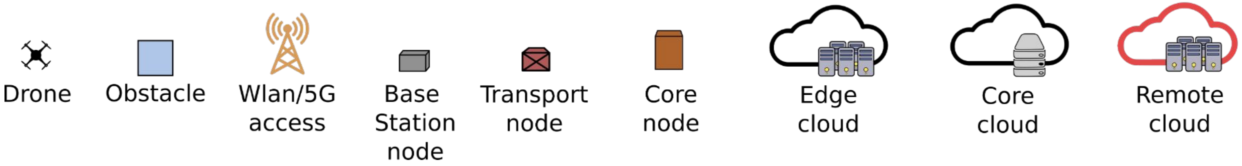
UAV Scenario



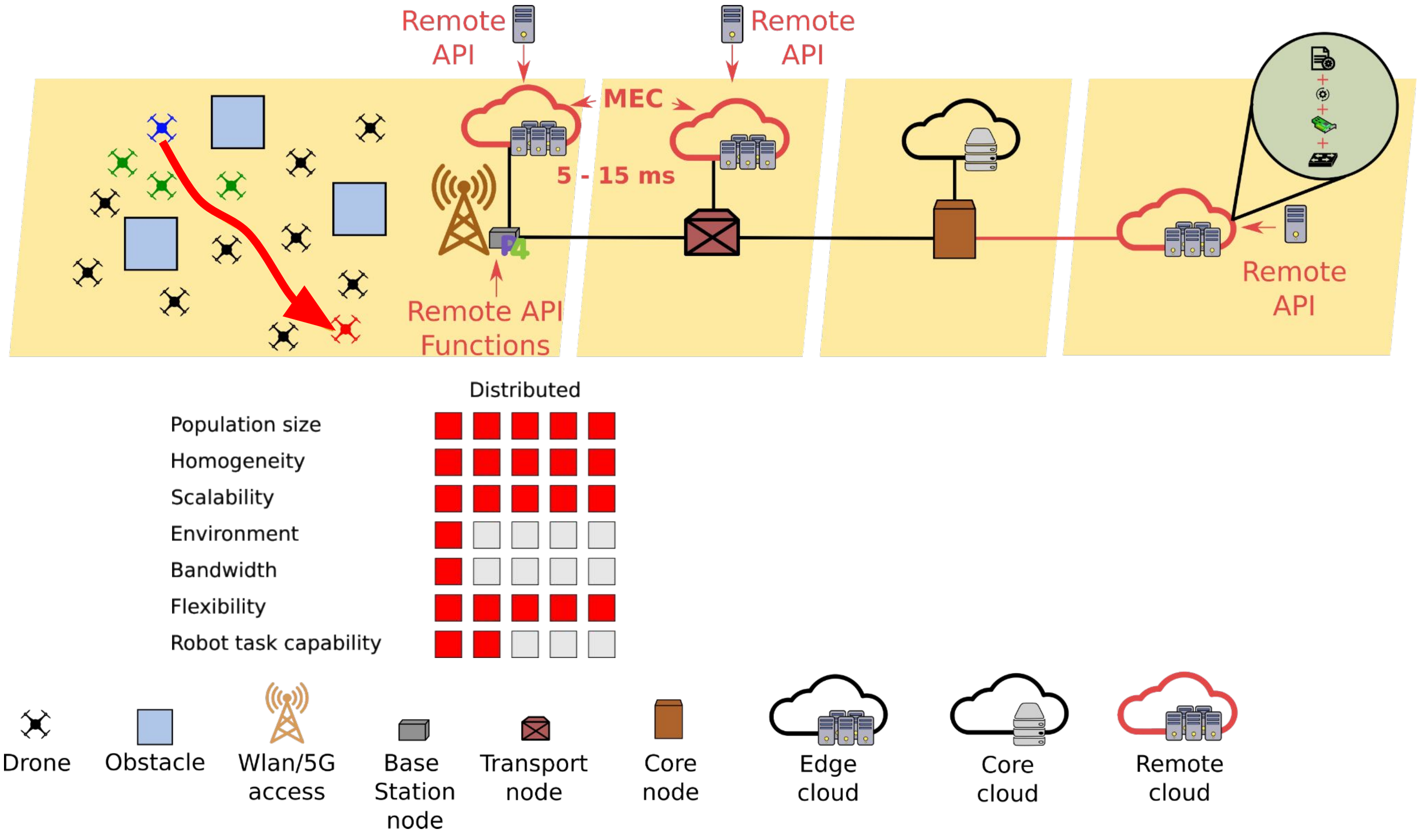


UAV Scenario

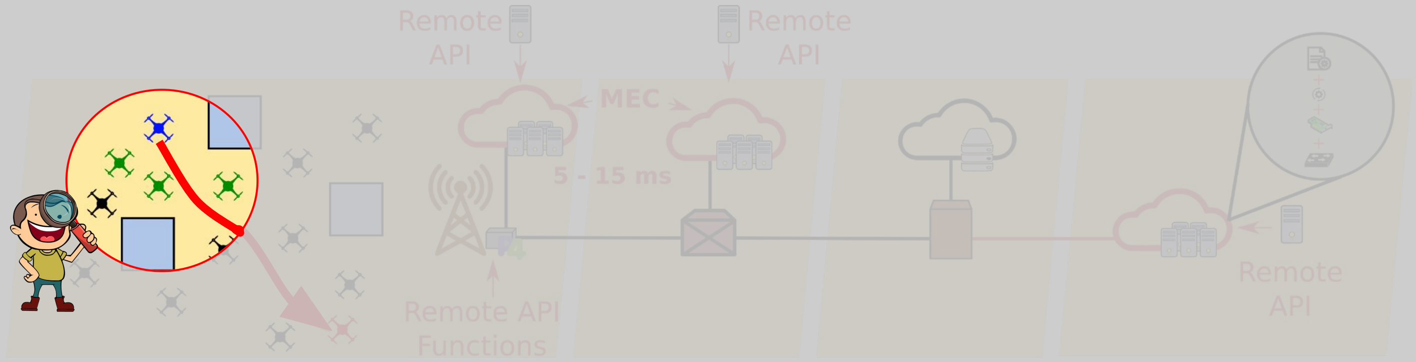




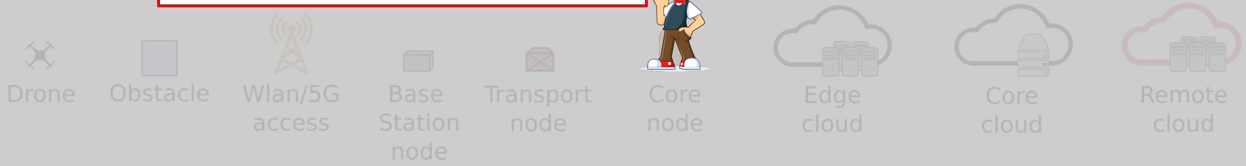
UAV Scenario



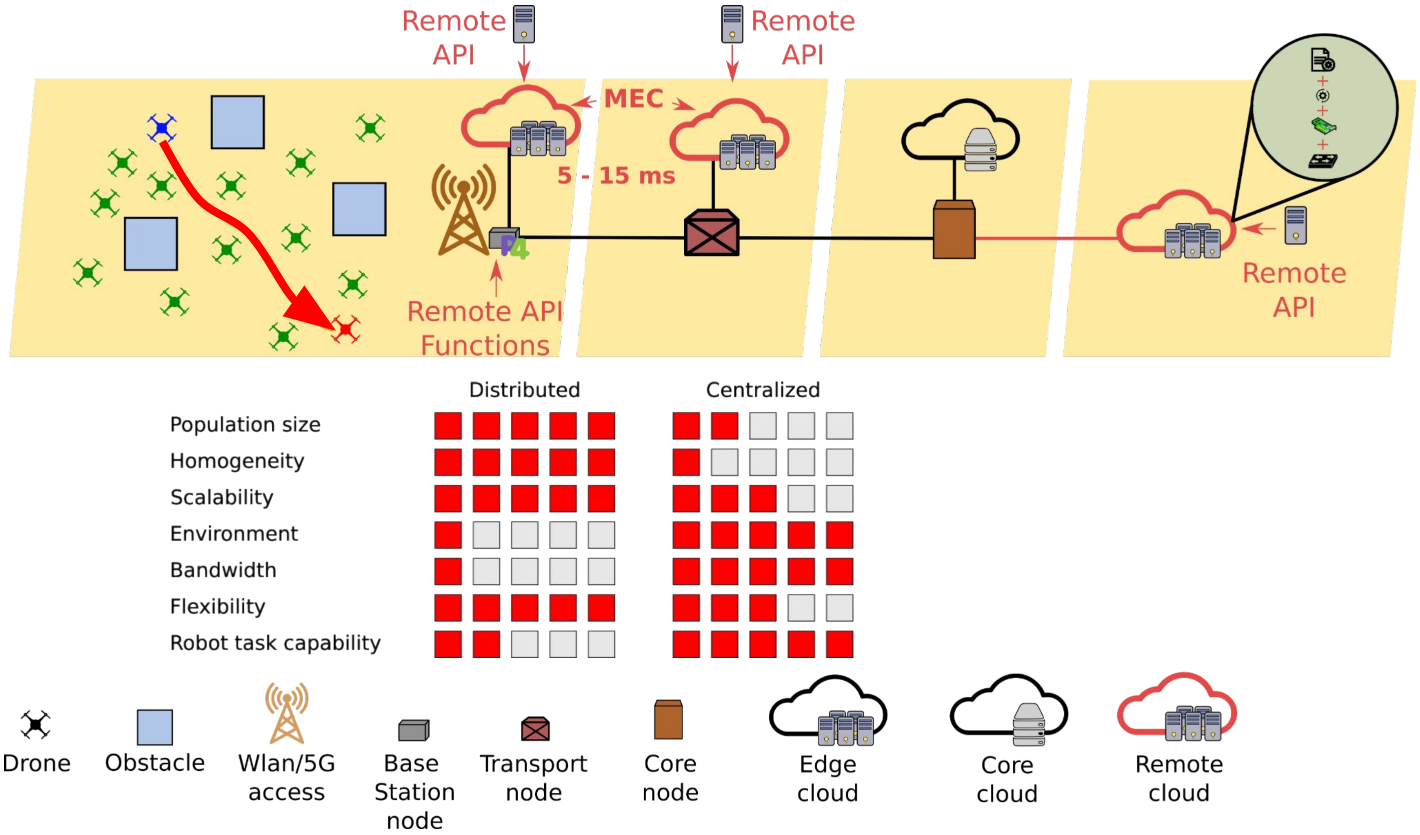
UAV Scenario



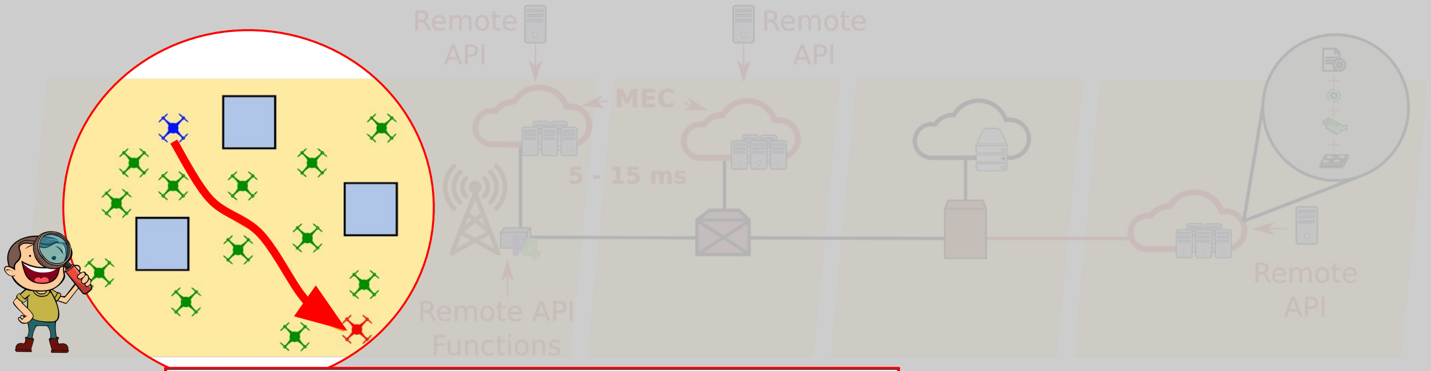
	Distributed
Population size	■ ■ ■ ■ ■
Homogeneity	■ ■ ■ ■ ■
Scalability	■ ■ ■ ■ ■
Environment	■ □ □ □ □
Bandwidth	■ □ □ □ □
Flexibility	■ ■ ■ ■ ■
Robot task capability	■ ■ □ □ □



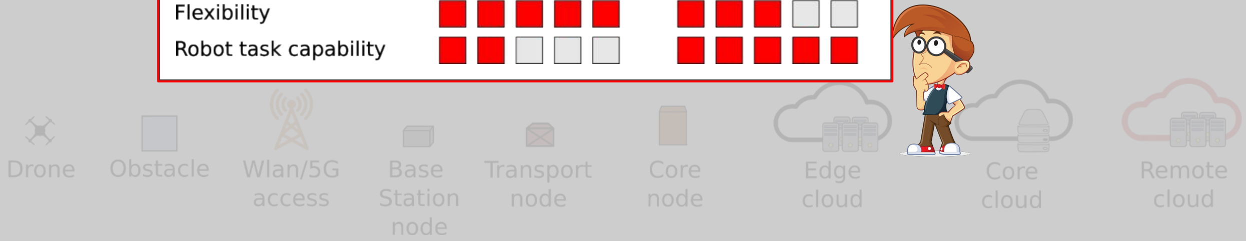
UAV Scenario



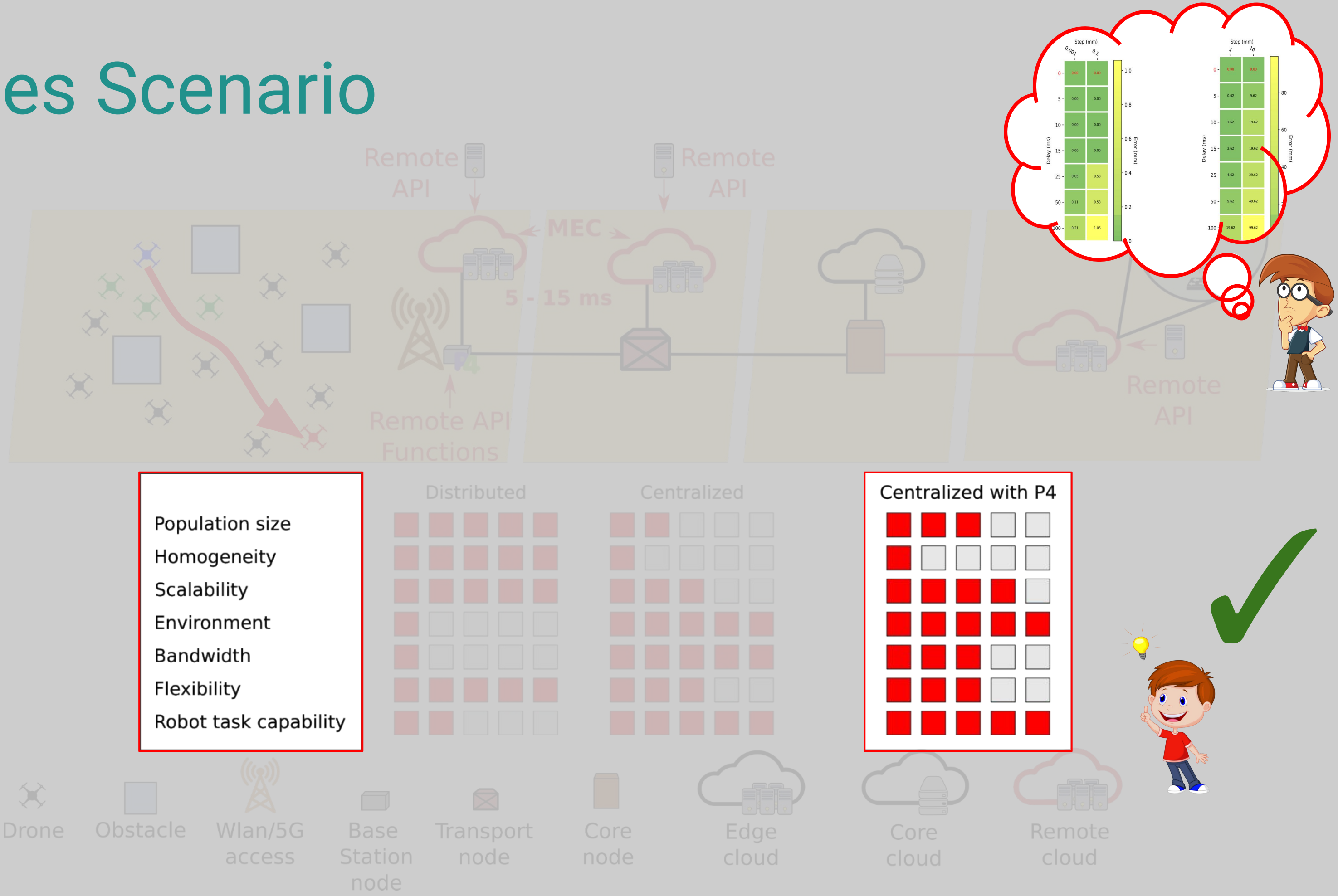
UAV Scenario



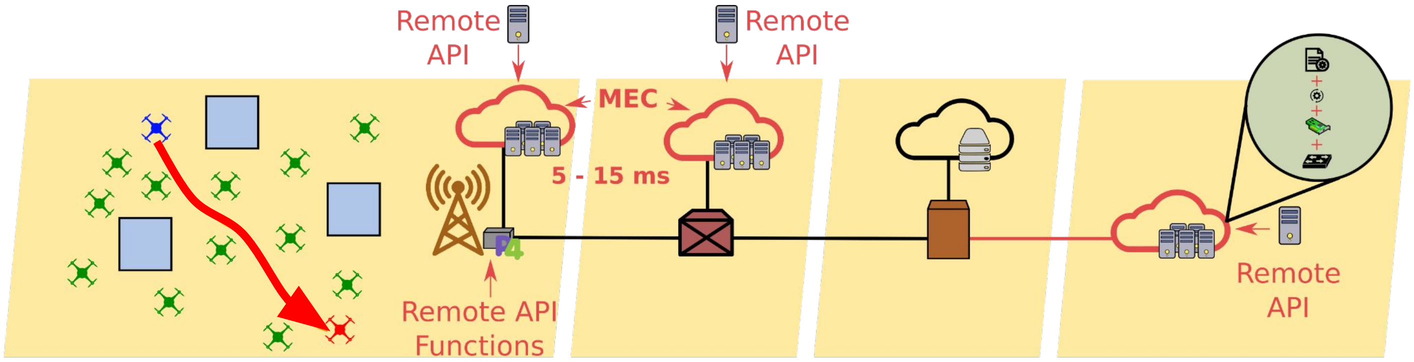
	Distributed	Centralized
Population size	■■■■■	■■□□□
Homogeneity	■■■■■	■□□□□
Scalability	■■■■■	■■■□□
Environment	■□□□□	■■■■■
Bandwidth	■□□□□	■■■■■
Flexibility	■■■■■	■■■□□
Robot task capability	■■□□□	■■■■■



Drones Scenario

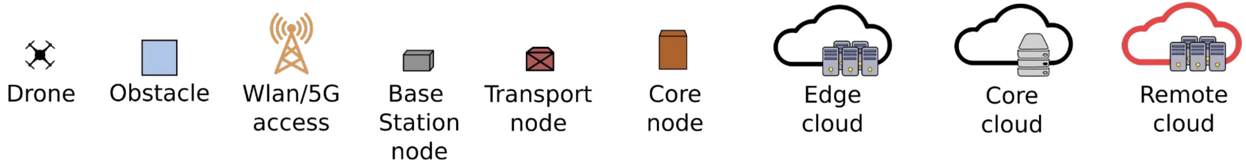


Drones Scenario

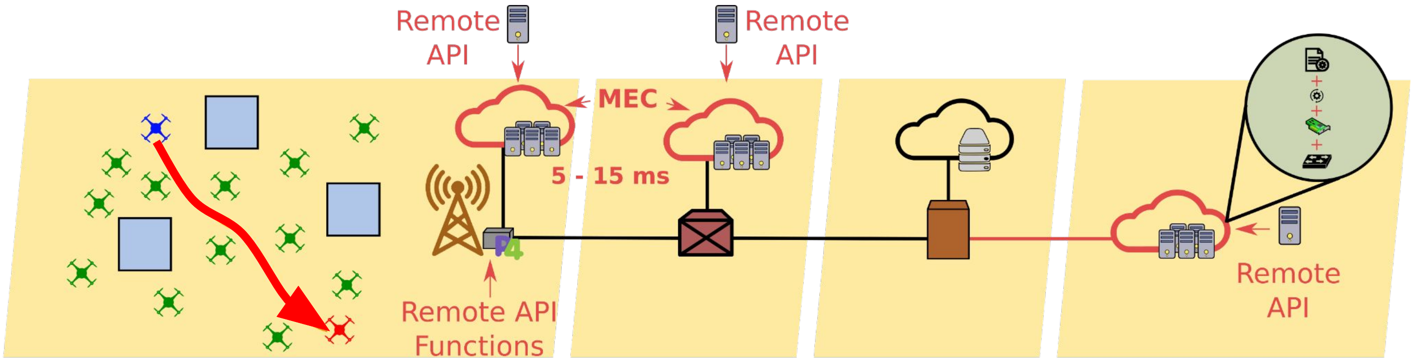


- High Performance
- Reconfigurability
- Protocol Independent
- Closer than a MEC approach

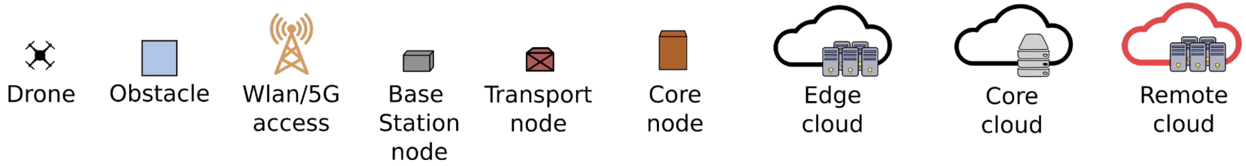
- Complex Operations
- Limited processing
- Limited memory



Drones Scenario



- Collision avoidance algorithm



Collision Avoidance Control Strategies

Geometric guidance	Best suitable for collision avoidance in the multi-vehicle system. It is not necessary to do extensive predictions and analysis (e.g., VFH+).

Collision cone	Set of vehicle velocities that will cause a collision with the obstacle if each of both the vehicle and obstacle is constant (e.g., VFH+, 3DVFH+).

Collision Avoidance Control Strategies

Geometric guidance	Best suitable for collision avoidance in the multi-vehicle system. It is not necessary to do extensive predictions and analysis (e.g., VFH+).
Collision cone	Set of vehicle velocities that will cause a collision with the obstacle if each of both the vehicle and obstacle is constant (e.g., VFH+, 3DVFH+).

Motion planning of teams of multi-UAVs	Geometric guidance is a local planning and may lead a deadlock, where more UAVs block each other in a way such that none of them is able to continue going to their corresponding goals.
Prioritized planning	Each UAV is given a priority according to its operation Fast in a complex environment, specifically if large teams of multi-UAVs

Collision Avoidance Control Strategies

Collision avoidance	**Path planning**	Generates a detailed trajectory for each vehicle to reach its final waypoint by using a cost.
	Conflict resolution	Multi-vehicle systems range widely in type and mission, it is not practical to require them depending on a ground station and predefined trajectories.
	Model predictive control	Simultaneously handling with different constraints and cost functions by optimization.
	Potential field	Design an attractive potential at the target and a repulsive potential is constructed for the obstacles.
	Geometric guidance	Best suitable for collision avoidance in the multi-vehicle system. It is not necessary to do extensive predictions and analysis (e.g., VFH+).
	Motion planning of teams of multi-UAVs	Geometric guidance is a local planning and may lead a deadlock, where more UAVs block each other in a way such that none of them is able to continue going to their corresponding goals.

In-network P4 function?

Complex algorithms (no suitable)
Possible dataplane implementation

Collision Avoidance Control Strategies

Collision avoidance	Path planning	Generates a detailed trajectory for each vehicle to reach its final waypoint by using a cost.
	Conflict resolution	Multi-vehicle systems range widely in type and mission, it is not practical to require them depending on a ground station and predefined trajectories.
	Model predictive control	Simultaneously handling with different constraints and cost functions by optimization.
	Potential field	Design an attractive potential at the target and a repulsive potential is constructed for the obstacles.
	Geometric guidance	Best suitable for collision avoidance in the multi-vehicle system. It is not necessary to do extensive predictions and analysis (e.g., VFH+).
	Motion planning of teams of multi-UAVs	Geometric guidance is a local planning and may lead a deadlock, where more UAVs block each other in a way such that none of them is able to continue going to their corresponding goals.

In-network P4 function? ➔

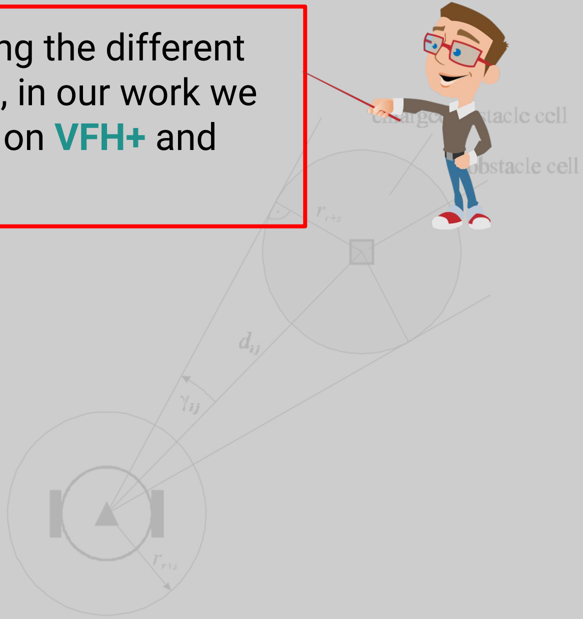
Complex algorithms (no suitable)
Possible dataplane implementation

Collision Avoidance Control Strategies

- **VFH+** is a collision cone method suitable for a P4 implementation due to its characteristics.
- Smoother robot trajectories and greater reliability.
- Map grid (histogram).
- Four-stage data reduction process to compute the new direction of motion.

- In parallel, **3DVFH+** is an enhanced version of VFH+ that add a 3D vision to the algorithm.
- The algorithm is able to respond in average of 300 µs.

Considering the different strategies, in our work we will focus on **VFH+** and **3DVFH+**.

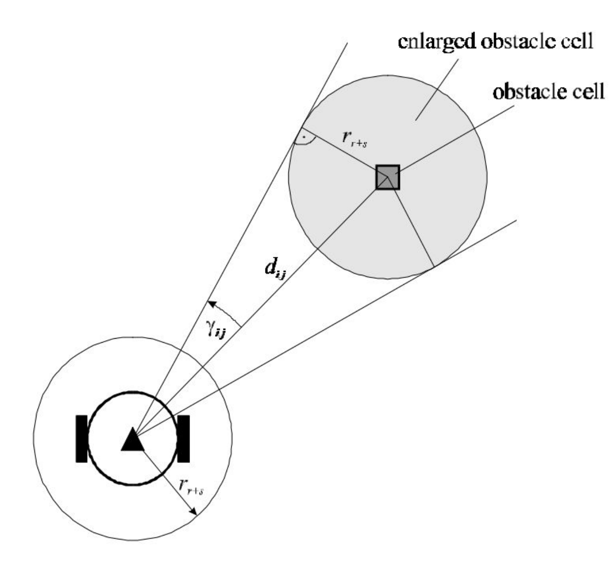


Example of VFH+ collision cone

Collision Avoidance Control Strategies

- **VFH+** is a collision cone method suitable for a P4 implementation due to its characteristics.
- Smoother robot trajectories and greater reliability.
- Map grid (histogram).
- Four-stage data reduction process to compute the new direction of motion.

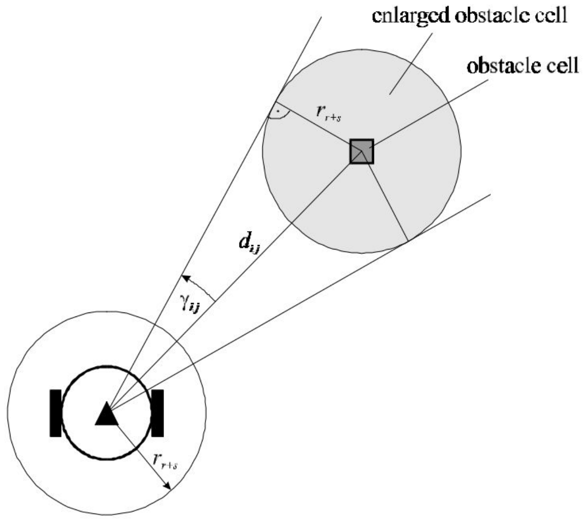
- In parallel, **3DVFH+** is an enhanced version of VFH+ that add a 3D vision to the algorithm.
- The algorithm is able to respond in average of 300 µs.



Example of VFH+ collision cone

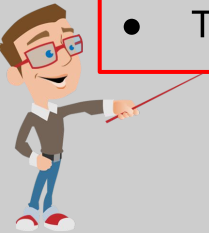
Collision Avoidance Control Strategies

- **VFH+** is a collision cone method suitable for a P4 implementation due to its characteristics.
- Smoother robot trajectories and greater reliability.
- Map grid (histogram).
- Four-stage data reduction process to compute the new direction of motion.



Example of VFH+ collision cone

- In parallel, **3DVFH+** is an enhanced version of VFH+ that add a 3D vision to the algorithm.
- The algorithm is able to respond in average of 300 µs.



Schedule and Next Steps

		2018		2019		2020		2021		2022		2023	
Activities		**-**	**S2**	**S1**	**S2**	**S1**	**S2**	**S1**	**S2**	**S1**	**S2**	**S1**	**S2**
01	Literature Review Industrial Robots												
02	Robot Arm + P4												
03	Literature review Collision avoidance												
04	UAV + Remote API												
05	UAV + P4												
06	Network characteristics emulation												
07	P4 vs Remote API												
08	Performance Evaluation												
09	Writing of Papers												
10	Thesis Writing												
11	Thesis Defense												

Green - Completed, Blue - In progress, Red - Todo

In-network applications related work

Scope	Cite	Aproach	Evaluation	Platform	Main contribution
Industrial networking	[95]	Monitoring, control and caching system	bmv2	S	reduce delay
	[49]	Coordinate transformation	Tofino	H	latency-critical
	[27]	Convolution filters	SmartNIC	H	line-following algorithm
	[78]	BPF programs	SmartNICs	H	reduce delay
	[41]	Querry processing	tofino/bmv2	H/S	Linear Road benchmark
	[80]	Data aggregation	bmv2	S	data reduction
	[52]	P4 primitives	bmv2	S	DDoS attack detection
	[47]	Event detection and state machine	bmv2	S	CEP
	[59]	-	Unity-based	S	reduce latency
	[101]	Packet Classification and queuing	tofino	H	P4-enabled network slicing
	[a]	Data parser and registers	tofino/bmv2	H/S	Collision avoidance algorithm

[a] Our proposed solution

Network emulation related work

Scope	Cite	Usability	Platform	Basic link metrics	Advance link metrics	Custom P4 support
Network Emulator	[51]	● ● ●	SW	Yes	Yes	bmv2
	[48]	● ○ ○	SW	No	No	No
	[55]	● ○ ○	SW	No	No	bmv2
	[42]	● ○ ○	SW	Yes	No	No
	[53]	● ○ ○	SW	Yes	No	Yes
	[17]	● ● ○	HW	No	Yes	No
	P7[a]	● ● ●	HW	Yes	Yes	Yes

[a]Our proposed solution

Context

5G Networks

With next-generation 5G networks just around the corner, new opportunities and challenges appear.

Industry 4.0

Applications in different trends (e.g., industry 4.0, agriculture, IoT, UAV) are gaining new envisions.

Requirements

Having high amounts of transmitted data and various network services, the network's capabilities need to grow together.

Motivation

- Ultra-low latency (ULL) applications with extremely low loss and delay variation are required in new generation networks.
- Horizontal (e.g., distance, nodes processing) and vertical (e.g., NIC, OS, Hypervisor, Application) delays affect the performance of end-to-end communication.
- Unmanned aerial vehicles (UAVs), specially Quadcopters are becoming an important application powered by the new network characteristics.

Problem statement

- Having external sensors (e.g., camera, GPS, proximity sensor), UAVs can observe sudden changes in the operational area (e.g., unexpected obstacles, other UAVs).
- In critical circumstances, not reacting in time (e.g., adjusting movements) can entail a severe impact, including collision, damages, or, more importantly, human injuries.
- To resolve this, each robot is directly connected to a powerful Remote API for continuous monitoring (e.g., complete view of the environment) and swift and precise interventions.

Challenges

Fast response

Reacting in time to avoid accidents (ULL application).

In-network functions

To reduce the response time, the functions (e.g., collision avoidance algorithm) can be deployed in the edge device.

Data Plane

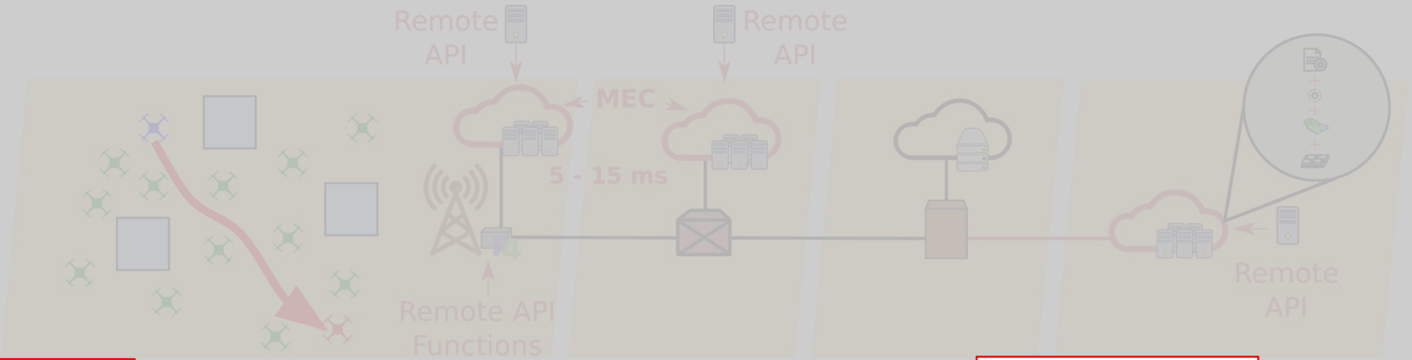
Considering the possible hardware limitations, the functions must respond as if they were deployed in the remote API.

Solution

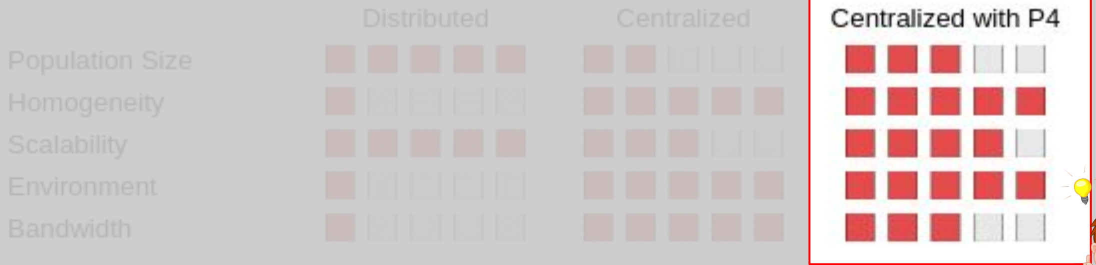
In-network Centralized Collision Avoidance Algorithm in Programmable Data Planes

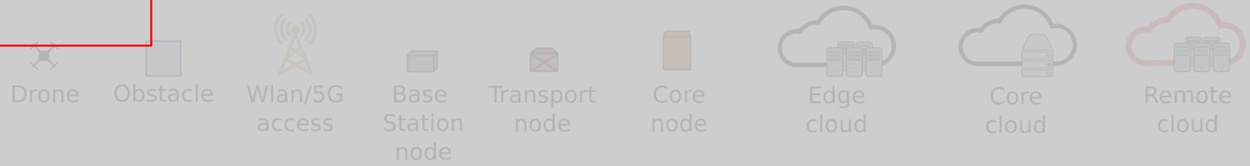
We propose bringing in-network functions to edge Programmable Data Planes, exploding the benefits of a P4 device (e.g., High Performance, Reconfigurability, Protocol Independent).

Quadcopter Scenario

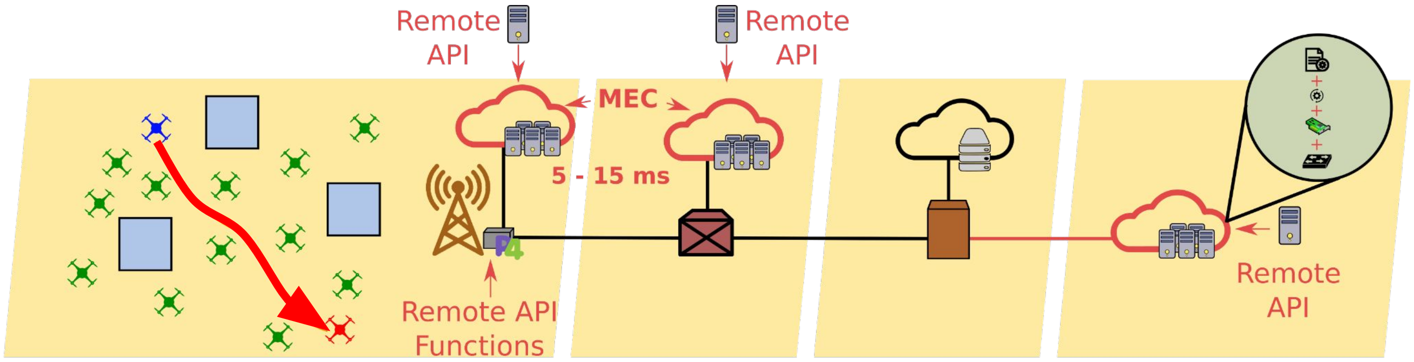


A centralized environment with remote API functions in the edge P4 device can empower the control.

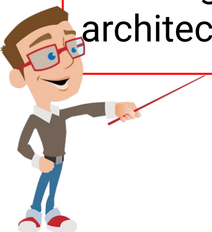




Quadcopter Scenario

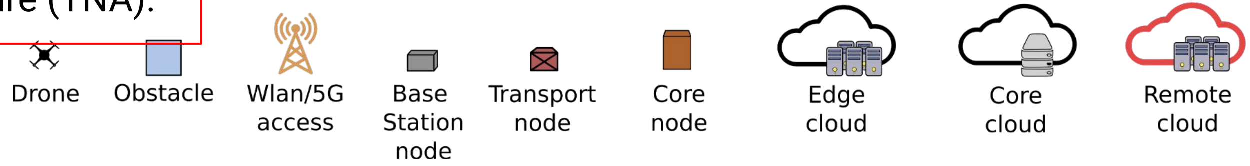


The edge P4 device has several benefits, but we must consider the challenges of the architecture (TNA).

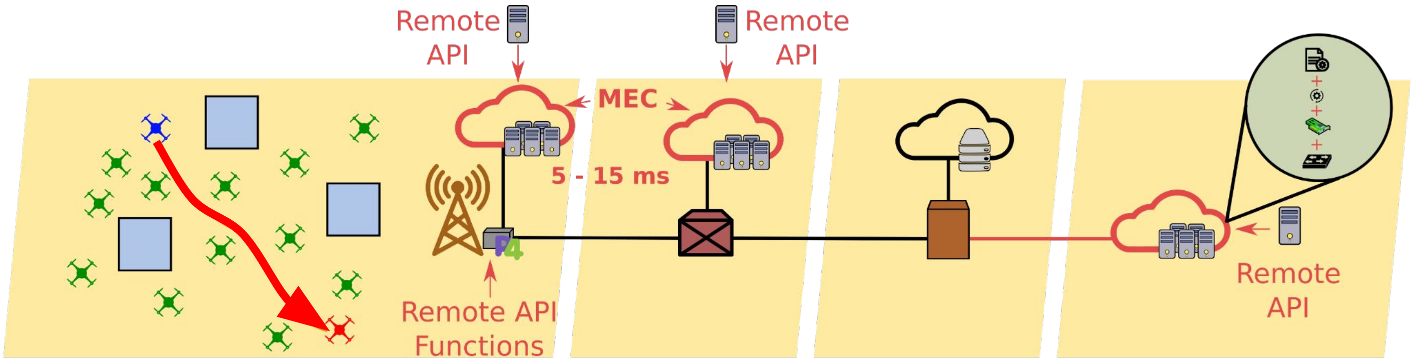


- High Performance
- Reconfigurability
- Protocol Independent
- Closer than a MEC approach

- Complex Operations
- Limited processing
- Limited memory

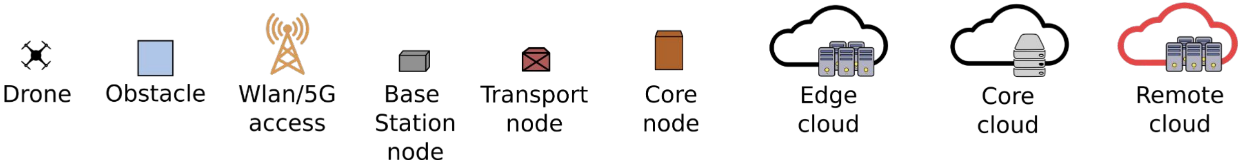


Quadcopter Scenario



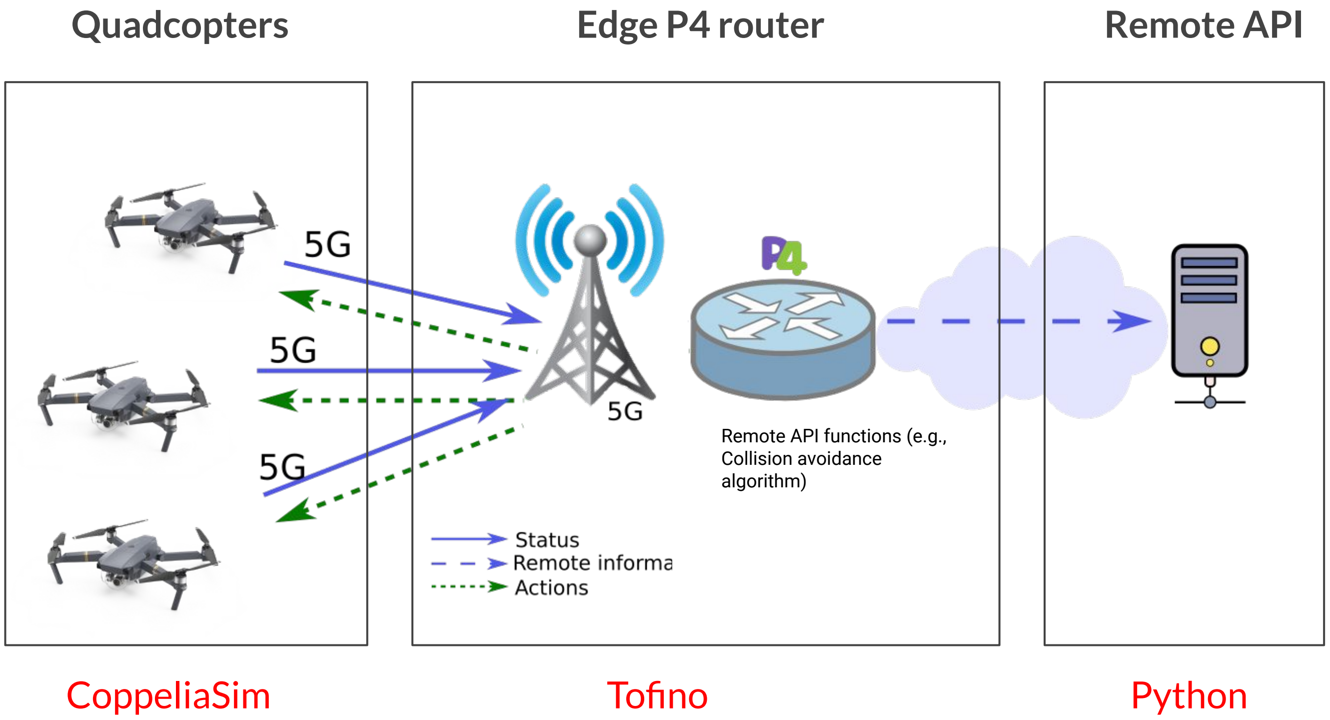
With these considerations, some of the remote API functions can be implemented and deployed in a P4 device.

- **Collision avoidance algorithm**
- Path planning algorithm



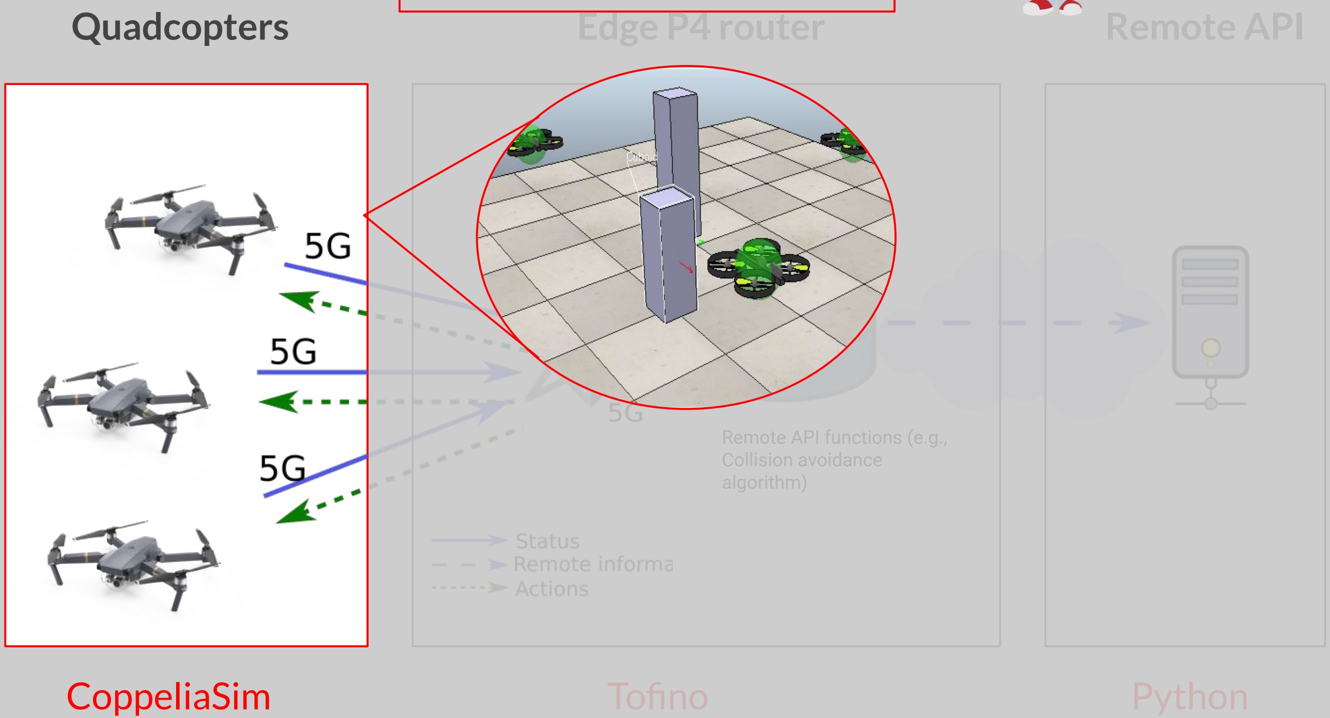
Implementation

Testbed


Quadcopters
Edge P4 router
Remote API
5G
5G
5G
5G
P4
Remote API functions (e.g.,
Collision avoidance
algorithm)
Status
Remote informa
Actions
CoppeliaSim
Tofino
Python

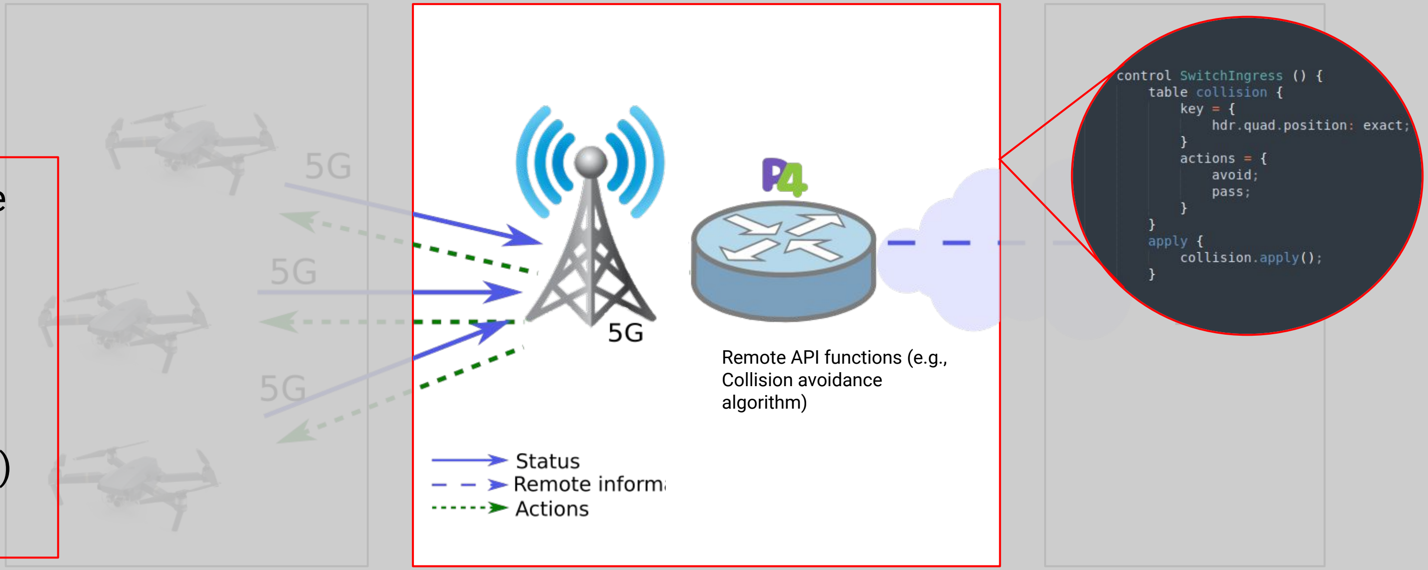
Testbed

Quadcopters are implemented in CoppeliaSim together with the flight mechanisms and hardware representations.



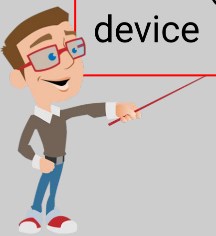
Testbed

Quadcopters | **Edge P4 router** | Remote API



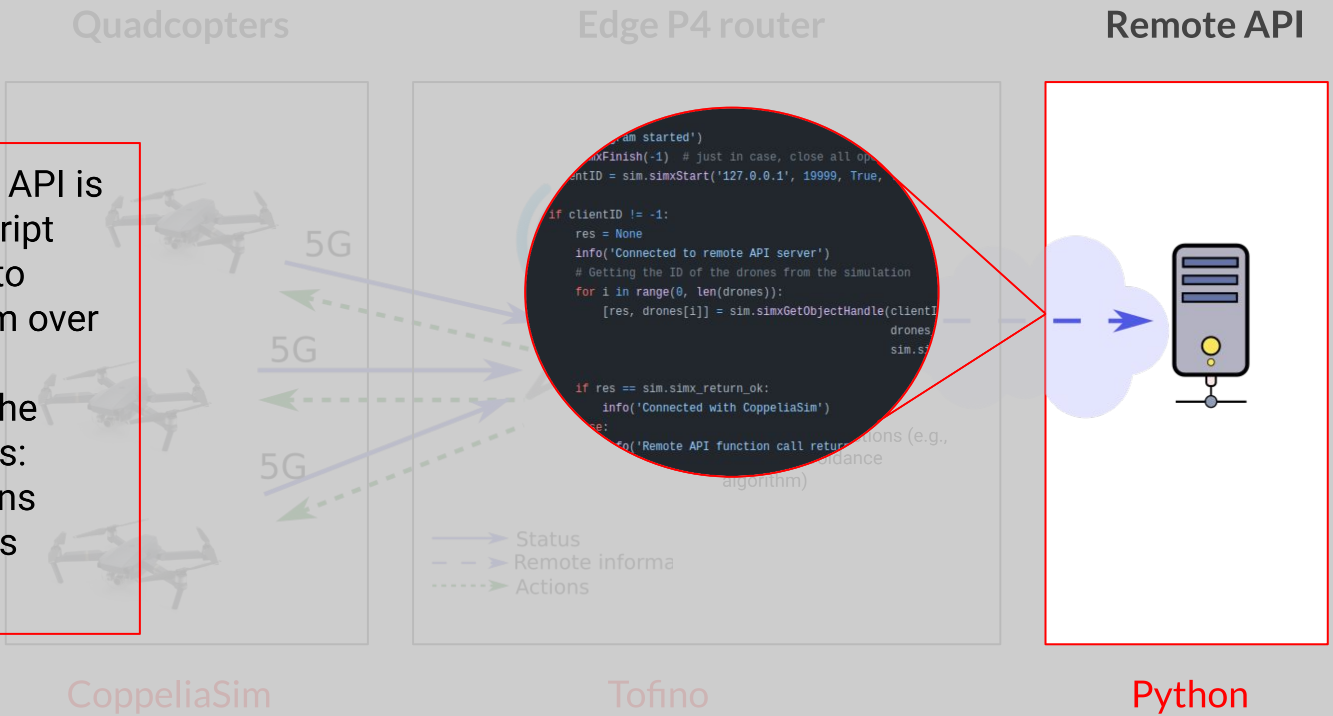
CoppeliaSim | Tofino | Python

The P4 code with the collision avoidance algorithm runs in a Tofino (TNA) device



Testbed

The remote API is a python script connected to CoppeliaSim over TCP.
It controls the quadcopters:
- Destinations
- Formations
- Actions

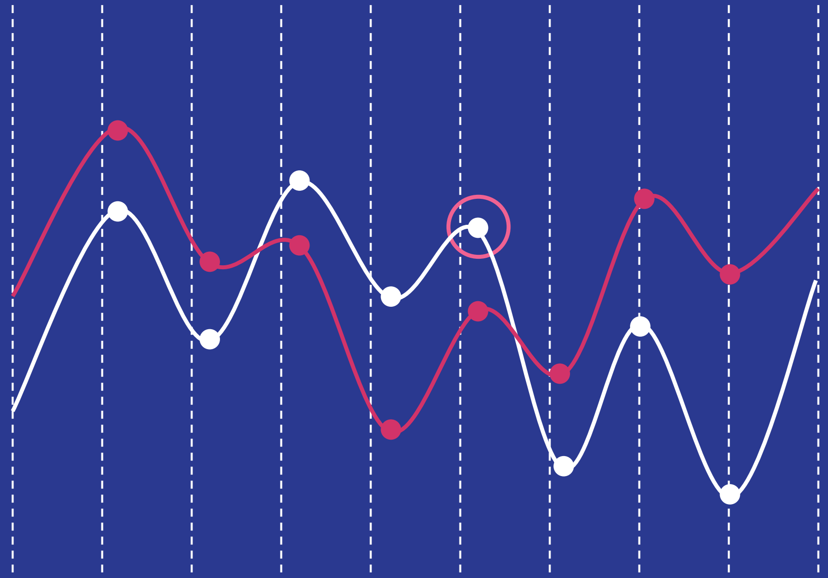


Required IT infrastructure

- Laptop.
- Remote P4 hardware device (i.e., tofino).
- Power.
- Internet connectivity.
- One monitor.

Results

Microbursts (generating)



References

[1] Brown, Gabriel. "New Transport Network Architectures for 5G RAN". Heavy Reading. 2018.

[2] Sutton, Andy. "5G Network Architecture, Design and Optimisation". 5G - State of Play. 2018.

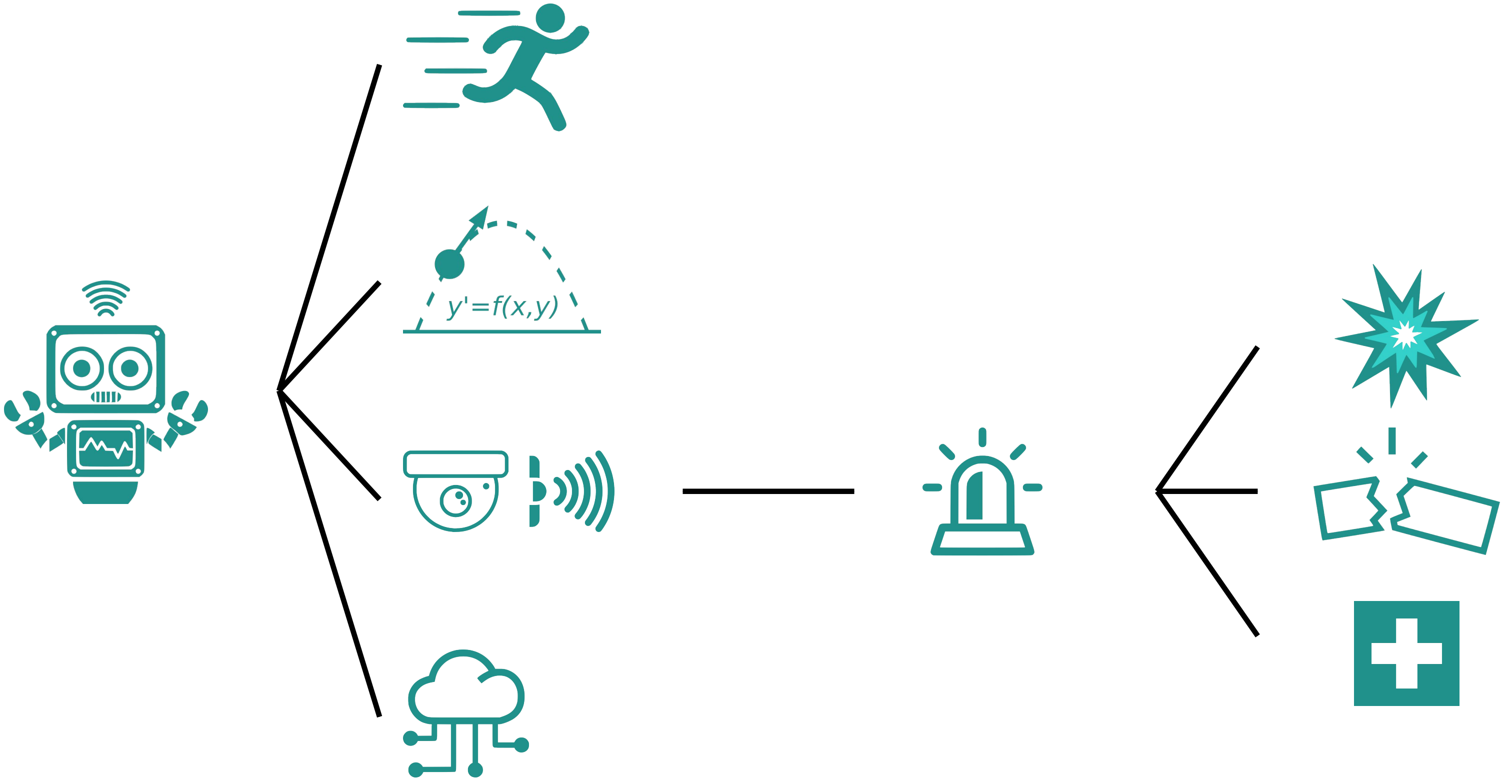
[3] Camps, Daniel. "Impact of 5G RAN Architecture in Transport Networks". ONDM - Optical Technologies in the 5G era. 2018.

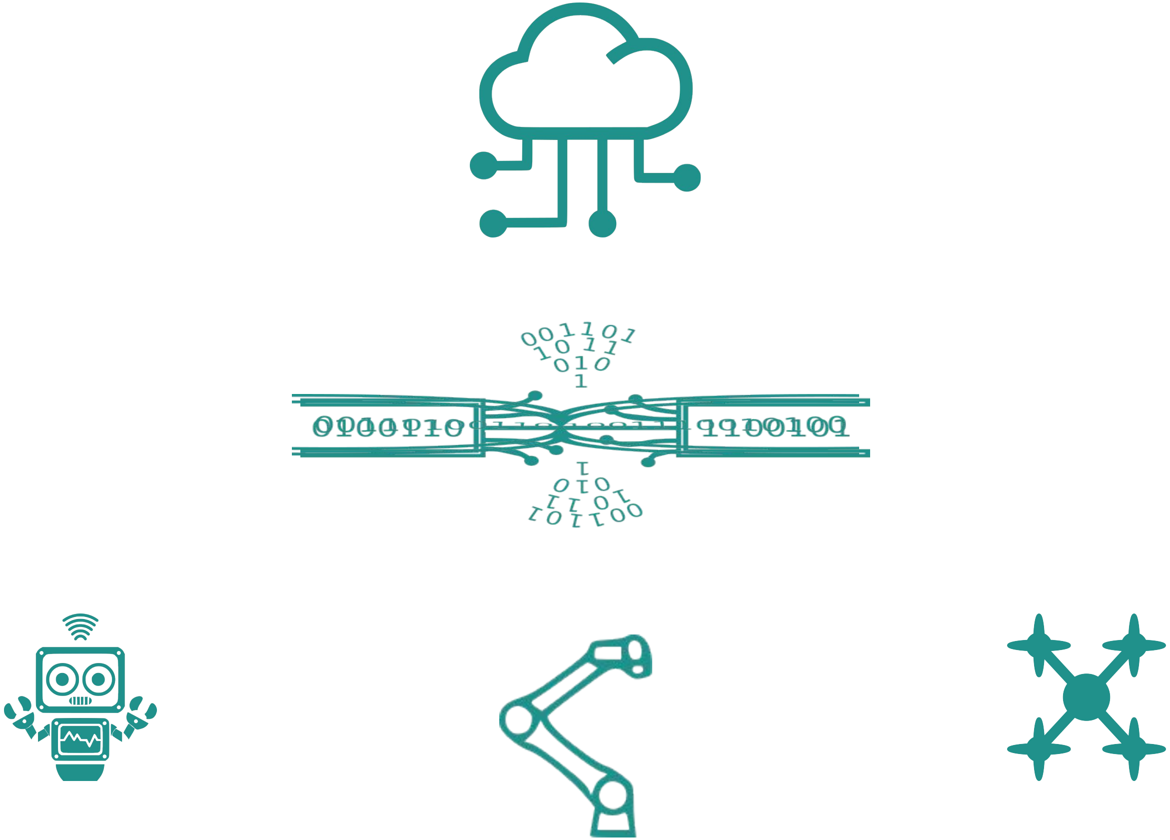
[4] Shah, S. et al. "SDN Enhanced Multi-Access Edge Computing (MEC) for E2E Mobility and QoS Management". IEEE Access. 2020.

[5] S. Das and M. Ruffini, "PON Virtualisation with EAST-WEST Communications for Low-Latency Converged Multi-Access Edge Computing (MEC)". OFC. 2020.

[6] Nikaein, N. et al. "LL-MEC: Enabling Low Latency Edge Applications". CloudNet. 2018.

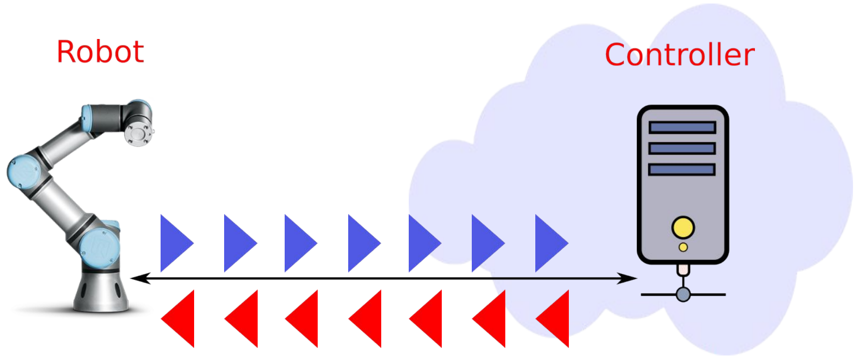
[7] Yuan, Q. et al. "Toward Latency-Optimal Placement and Autoscaling of Monitoring Functions in MEC". IEEE Access. 2020.


y'=f(x,y)

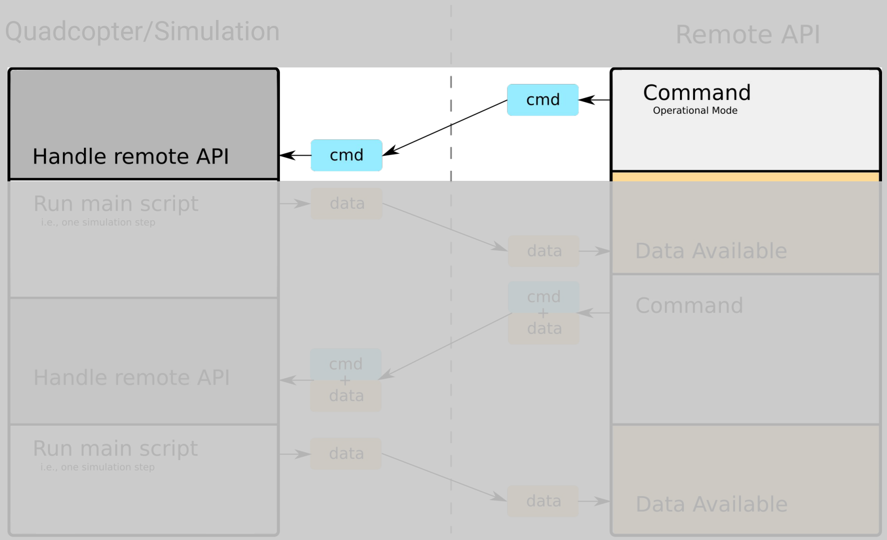


Introduction

Traditional robot control

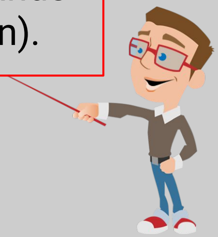


Quadcopter ⇒ Remote API

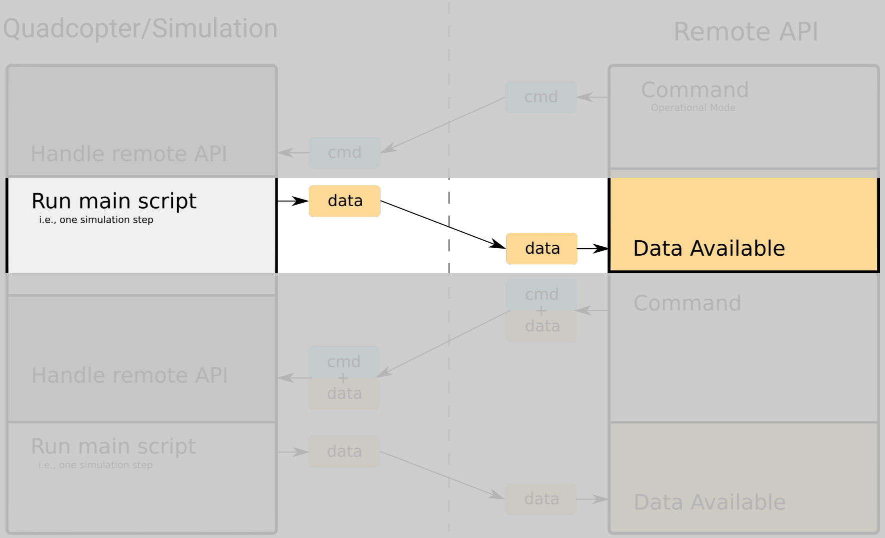


ID	Command		
ID	Position	Velocity	Sensor Data
ID	Command	Data	

The remote API can send commands to the simulator. (e.g., get position).

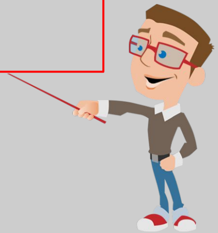


Quadcopter ⇒ Remote API

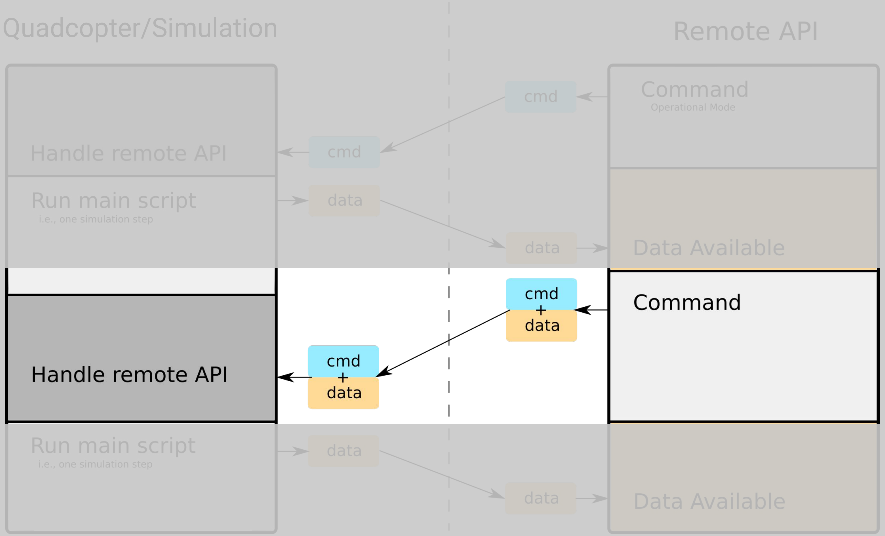


ID	Command		
ID	Position	Velocity	Sensor Data
ID	Command	Data	

The simulator process the commands and respond with the required data (e.g., current position).



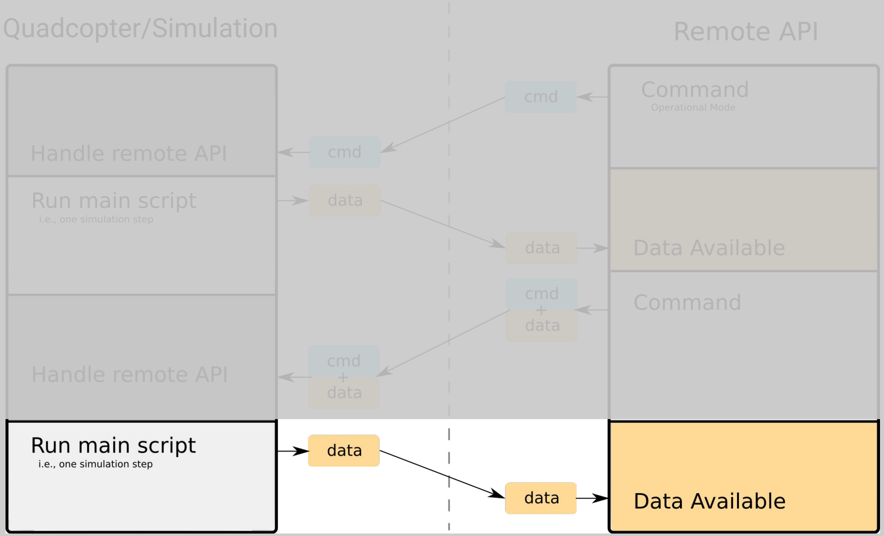
Quadcopter ⇒ Remote API

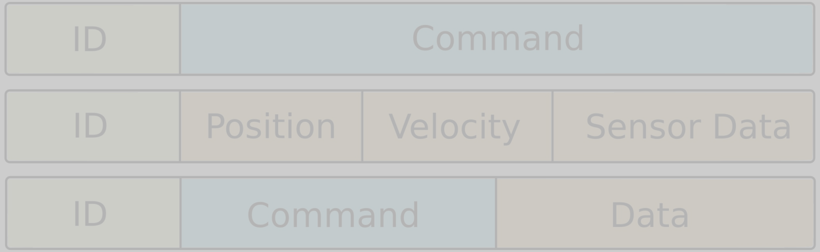


ID	Command		
ID	Position	Velocity	Sensor Data
ID	Command	Data	

The commands can contain additional data (e.g., set position).

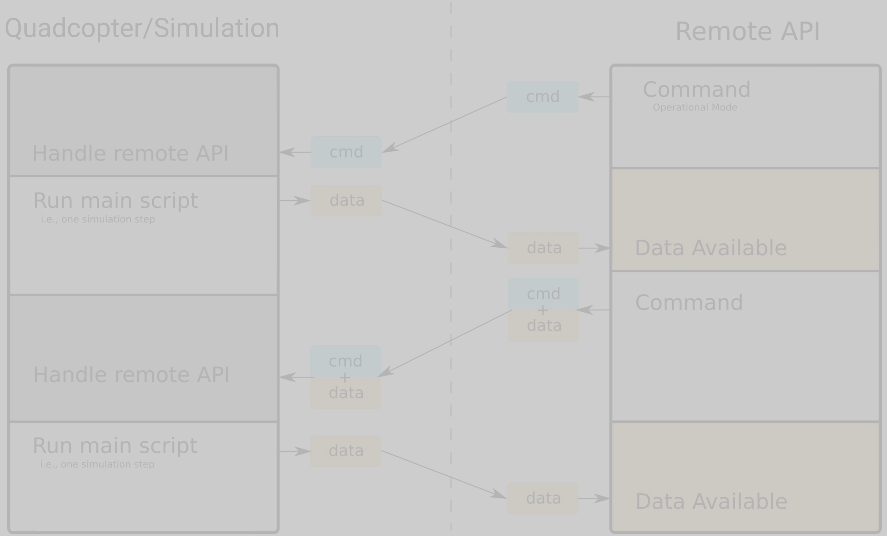
Quadcopter ⇒ Remote API





The simulator processes the commands and responds with an acknowledgment.

Quadcopter ⇒ Remote API



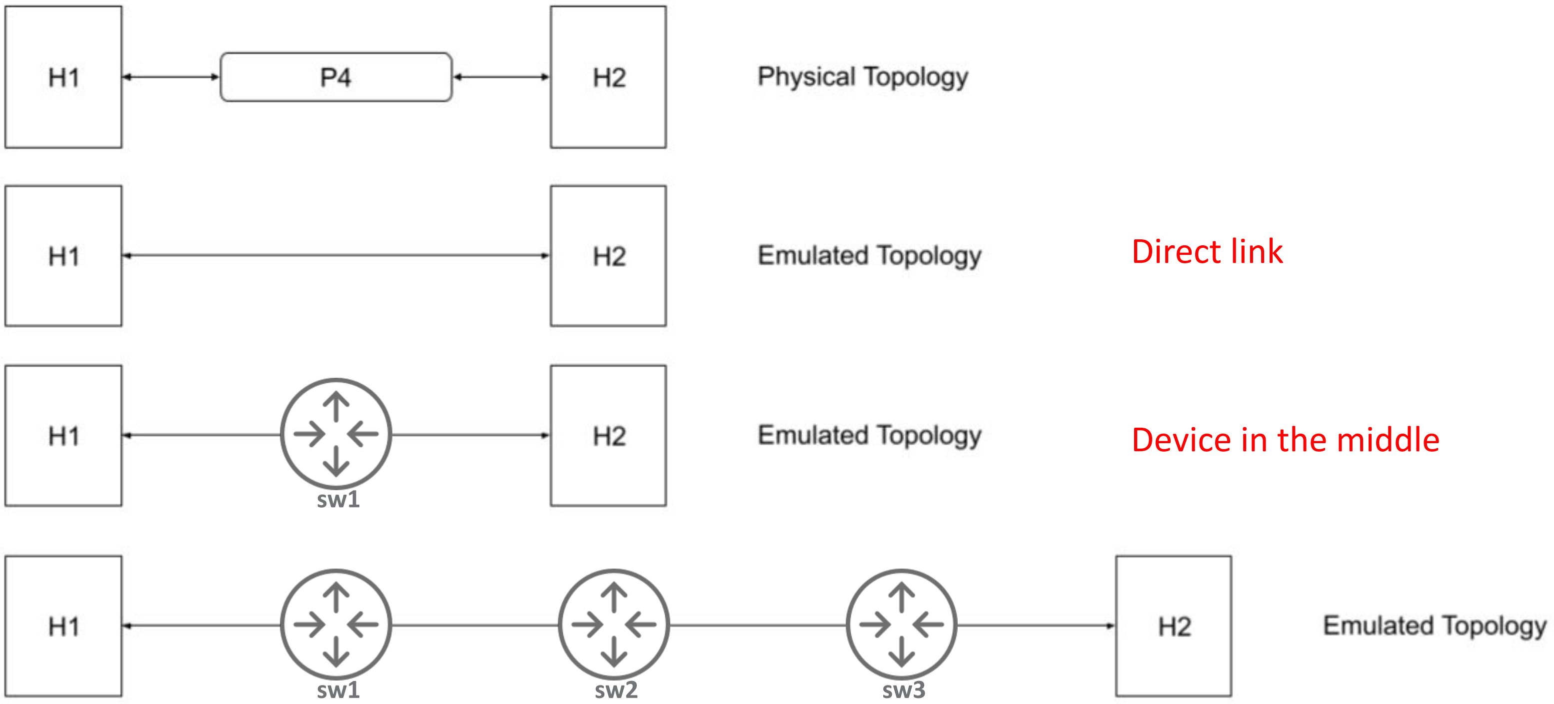
ID	Command

ID	Position	Velocity	Sensor Data

ID	Command	Data

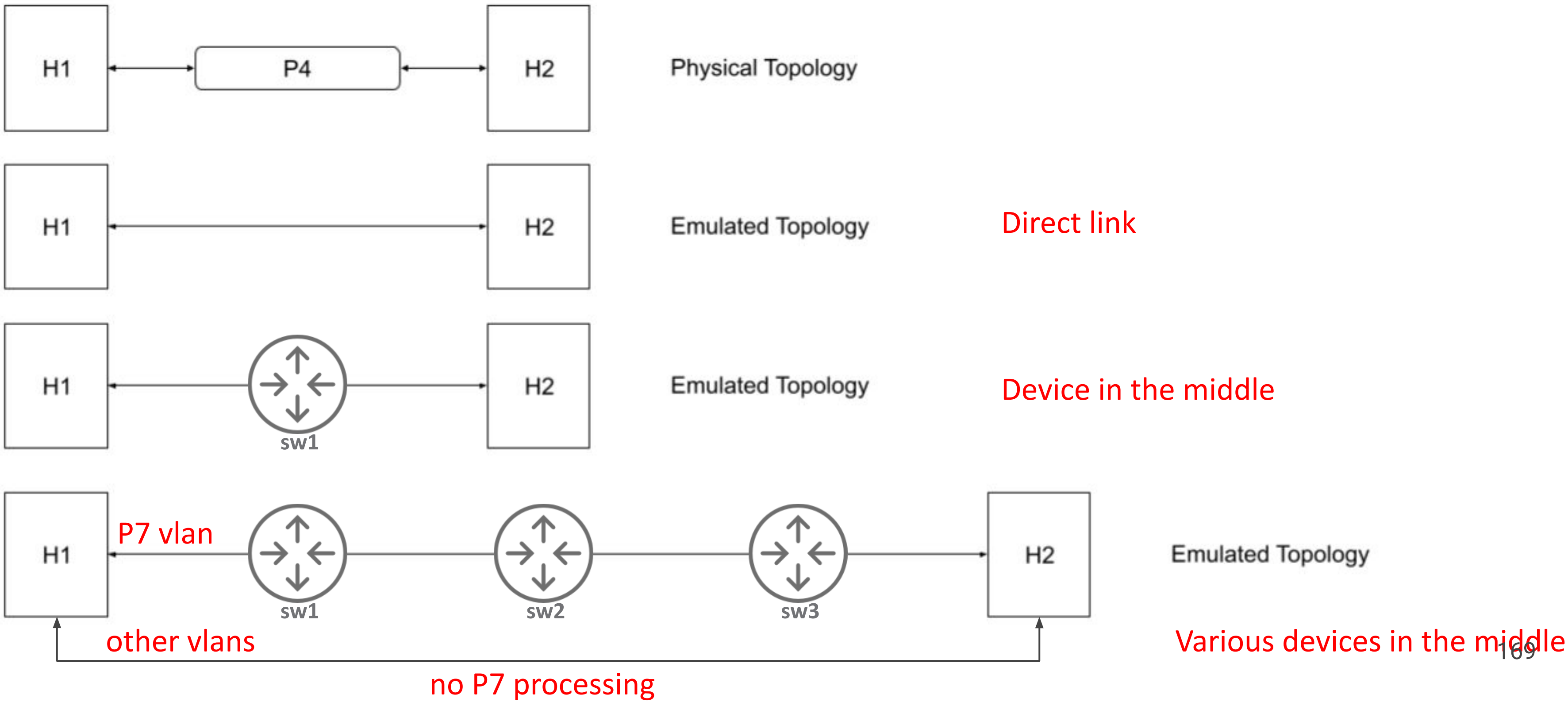
The command and the data response have a specific packet structure.

Link characteristics



Various devices in the middle

Link characteristics



Direct link

```
from src.data import *

topo = generator('main')

# Stratum ip:port
topo.addstratum("192.168.110.238:9559")

# Recirculation port default 68
topo.addrec_port(68)

# addhost(name,port,D_P,speed_bps,AU,FEC,vlan)
topo.addhost("h1",19, 20, 100000000000, "False", "False", 1920)
topo.addhost("h2",20, 28, 100000000000, "False", "False", 1920)

# addlink(node1, node2, bw, pkt_loss, latency)
topo.addlink("h1","h2", 100000000000, 0, 5)

topo.generate_chassis()
topo.generate_p4rt()
topo.generate_p4code()

topo.generate_graph()
```

Host name

Port number

Port speed

Port config

P7 vlan

Direct link

```
from src.data import *

topo = generator('main')

# Stratum ip:port
topo.addstratum("192.168.110.238:9559")

# Recirculation port default 68
topo.addrec_port(68)

# addhost(name,port,D_P,speed_bps,AU,FEC,vlan)
topo.addhost("h1",19, 20, 100000000000, "False", "False", 1920)
topo.addhost("h2",20, 28, 100000000000, "False", "False", 1920)

# addlink(node1, node2, bw, pkt_loss, latency)
topo.addlink("h1","h2", 100000000000, 0, 5)

topo.generate_chassis()
topo.generate_p4rt()
topo.generate_p4code()

topo.generate_graph()
```

node 1
node 2
link bw
latency
pkt loss

Device in the middle

```
from src.data import *

topo = generator('main')

# Stratum ip:port
topo.addstratum("192.168.110.238:9559")

# Recirculation port default 68
topo.addrec_port(68)

# addswitch(name)
topo.addswitch("sw1")

# addhost(name,port,D_P,speed_bps,AU,FEC,vlan)
# include the link configuration
topo.addhost("h1",19, 20, 100000000000, "False", "False", 1920)
topo.addhost("h2",20, 28, 100000000000, "False", "False", 1920)

# addlink(node1, node2, bw, pkt_loss, latency)
topo.addlink("h1","sw1", 100000000000, 0, 5)
topo.addlink("sw1","h2", 100000000000, 0, 5)

topo.generate_chassis()
topo.generate_p4rt()
topo.generate_p4code()

topo.generate_graph()
```

switch name

links