In-network P4-based Low Latency Robot Arm Control

Fabricio Rodriguez
frodri@dca.fee.unicamp.br
University of Campinas (UNICAMP)

Christian Esteve Rothenberg
chesteve@dca.fee.unicamp.br
University of Campinas (UNICAMP)

Gergely Pongrácz
gergely.pongracz@ericsson.com
Ericsson Research

ABSTRACT

Advances in Software Defined Networking data plane programmability are being fueled by new domain languages like P4, enabling to program the behavior of networking devices. At the crossroads, new types of applications in different verticals (e.g., industry 4.0, agriculture IoT, medicine) are developing new visionary footprints through the arrival of 5G networks. Many of the envisioned applications (e.g. automation control) require high reliability and very low latency with bounded guarantees. In-network hardware solutions defined in P4 can become enabling solutions to support these kind of applications. In this work, we present a strawman robot arm control application running inside a P4 switch capable of parsing robot arm position messages to timely craft a message within a TCP connection to send a stop command at very low latency scales.

CCS CONCEPTS

• **Networks** → **Programmable networks**; **In-network processing**; • **Computer systems organization** → *Robotic control.*

KEYWORDS

P4, Low Latency applications, Robotic control

ACM Reference Format:
Fabricio Rodriguez, Christian Esteve Rothenberg, and Gergely Pongrácz. 2019. In-network P4-based Low Latency Robot Arm Control. In *The 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '19 Companion), December 9–12, 2019, Orlando, FL, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3360468.3368178

1 INTRODUCTION

With the arrival of next-generation 5G networks, new opportunities and challenges appear. Having significantly increased amounts of transmitted data and diverse network services, the capabilities of the network have to improve conjointly. It becomes paramount to deliver end-to-end networks capable of supporting Ultra-low latency (ULL) applications with extremely low loss and delay variations, i.e., notable improvements in terms of Quality of Service (QoS) are required [2]. 5G encloses a wide range of applications, including personal-to-person, machine-type communications (MTC) with applications ranging from Internet-of-Things (IoT) sensing and monitoring, smart cites and homes, robotic control and industrial automation, to remote surgery [1].

The evolving Software Defined Networking (SDN) landscape through advances in data plane application domain languages like Protocol-Independent Packet Processors (P4) [4] offers new opportunities opportunity in user-defined logic to process the packets inside network datapath pipelines. Based on a match+action tables pipeline model, P4 gives the possibility to define new packet headers and processing actions based on arbitrary header field contents, stateful registers, and primitive atomic operations.

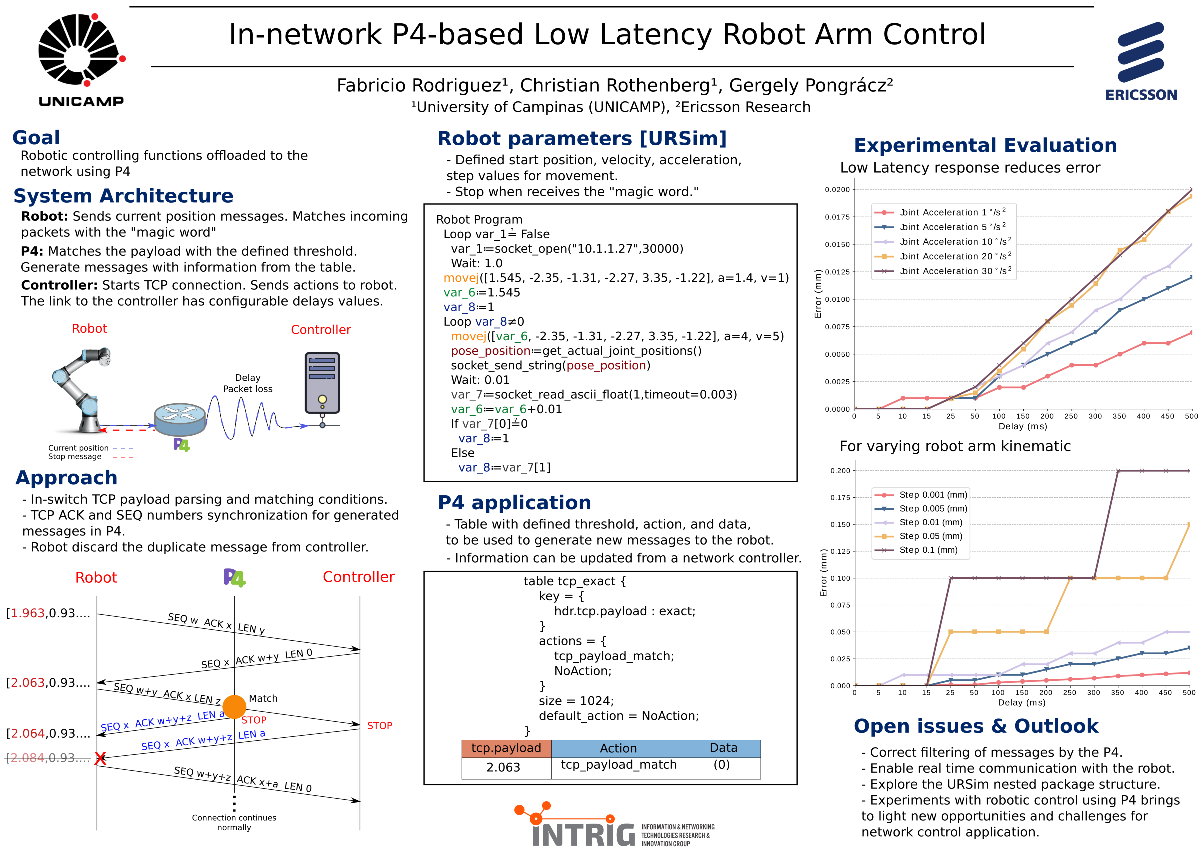
In this poster, we approach robotic control functions with respect to the possibility of being offloaded to programmable data plane devices performing in-network computations. Using P4, some basic tasks can be implemented to parse and extract the payload data, perform some calculations (e.g., distance, threshold, temperature), filtering, and much more, including crafting a custom messages sent to the robot or controller. The main advantages of an in-network P4 device approach are that all packet processing and operations can be done at line rate delivering very low latency due to physical proximity of edge P4 switches and bounded time guarantees due to hardware-based pipeline implementations.

To simulate the interactions and functions of robots, we use Universal Robots Simulation (URSim) [5], a widely used simulator for robot control programs and manual movement of robots. URSim allows to simulate the robot kinematic through digital inputs (I/O) and the definition of different metrics and parameters. URsim is able to communicate to a robot controller using TCP sockets.

2 SYSTEM OVERVIEW

One of the main goals of our first robotic control with P4 experiments is to investigate that, if we have a programmable device (e.g. P4 router) near to the target (Robot), it is possible to critically reduce the delay of vital actions (e.g., emergency stop, alarm notification). Another essential design objective is an in-network offloaded logic capable of analyzing the information transmitted from the target and taking actions (i.e., create or update messages).

Figure 1 presents a high-level overview of our system architecture and experiment setup. On the right side, we have the controller implementation as a Python script to start the TCP communication with the robot and analyze incoming messages to learn about the current robot arm position. On the left side, the robot is configured with a basic script (Robot parameters) to send the current position and read any incoming message from the controller. The robot is configured with a start position, velocity, acceleration, and step size for movements. A P4 router forwards the packets between the robot and the controller and analyzes (parses + matches) specific payload data (current robot 3D coordinates). For this in-network control application in P4, the match+action pipelines allows triggers the detection of specific robot position and the creation of correct TCP segment containing a reply message to the robot with a "magic word" command to immediately stop the robot movement.



In-network P4-based Low Latency Robot Arm Control

Fabricio Rodriguez¹, Christian Rothenberg¹, Gergely Pongrácz²
¹University of Campinas (UNICAMP), ²Ericsson Research

Goal
Robotic controlling functions offloaded to the network using P4

System Architecture
Robot: Sends current position messages. Matches incoming packets with the "magic word"
P4: Matches the payload with the defined threshold. Generate messages with information from the table.
Controller: Starts TCP connection. Sends actions to robot. The link to the controller has configurable delays values.

Approach
- In-switch TCP payload parsing and matching conditions.
- TCP ACK and SEQ numbers synchronization for generated messages in P4.
- Robot discard the duplicate message from controller.

Robot parameters [URSim]
- Defined start position, velocity, acceleration, step values for movement.
- Stop when receives the "magic word."

```
Robot Program
 Loop var_1≟ False
  var_1≔socket_open("10.1.1.27",30000)
  Wait: 1.0
 movej([1.545, -2.35, -1.31, -2.27, 3.35, -1.22], a=1.4, v=1)
 var_6≔1.545
 var_8≔1
 Loop var_8≠0
  movej([var_6, -2.35, -1.31, -2.27, 3.35, -1.22], a=4, v=5)
  pose_position≔get_actual_joint_positions()
  socket_send_string(pose_position)
  Wait: 0.01
  var_7≔socket_read_ascii_float(1,timeout=0.003)
  var_6≔var_6+0.01
  If var_7[0]≟0
   var_8≔1
  Else
   var_8≔var_7[1]
```

P4 application
- Table with defined threshold, action, and data, to be used to generate new messages to the robot.
- Information can be updated from a network controller.

```
table tcp_exact {
    key = {
        hdr.tcp.payload : exact;
    }
    actions = {
        tcp_payload_match;
        NoAction;
    }
    size = 1024;
    default_action = NoAction;
}
```

tcp.payload	Action	Data
2.063	tcp_payload_match	(0)

Experimental Evaluation
Low Latency response reduces error

Joint Acceleration 1 °/s²
Joint Acceleration 5 °/s²
Joint Acceleration 10 °/s²
Joint Acceleration 20 °/s²
Joint Acceleration 30 °/s²
Error (mm)
Delay (ms)
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
0 5 10 15 25 50 100 150 200 250 300 350 400 450 500

For varying robot arm kinematic

Open issues & Outlook
- Correct filtering of messages by the P4.
- Enable real time communication with the robot.
- Explore the URSim nested package structure.
- Experiments with robotic control using P4 brings to light new opportunities and challenges for network control application.

Figure 1: P4-based Low Latency Robot Arm Control, poster preview

One of the biggest challenges is to maintain the TCP session. If we manipulate or create any message in the P4 router, we need to adequately sync all the acknowledgment (ACK) and sequence (SEQ) numbers. To this end, we use an approach similar to TCP veto [3]. If the robot sends a position message that matches with the predefined threshold, the P4 router will forward the message to the robot controller and will create a new message (with the "magic word") to the robot with valid ACK and SEQ numbers. In this way, when the controller also sends the message, it will be discarded as duplicated, but the ACK and SEQ Numbers will be in sync.

3 PRELIMINARY RESULTS & FUTURE WORK

For this experimental test, we assume that the robot controller can be at different distances from the robot, inducing delay to the link between the P4 router and the controller. For the link between the robot and the P4, we assume that they are close, with direct link and very low latency.

In our proof of concept prototype implementation, we were able to verify the robotic control functions offloaded to the network. The P4 device correctly detects current position messages from the robot that matches with the defined threshold. Through P4 language constructs, new messages can be created with the information stored in the tables, being easy to update or modify it from a network controller. It is important to notice that all the messages created by the P4 need to have valid information, being crucial recalculate the checksum and update the ACK and SEQ numbers.

Our experimental evaluations demonstrate that the error (stop position difference) increases depending on the delay of the controller command. This error is mitigated when sending critical messages directly from the P4 device. The error depends on the kinematic configuration of the robot (i.e., steps of movement), marking a big difference when reducing the delay. With P4 actions directly in the network, latency can be reduced for critical use cases, opening opportunities for different range of applications e.g., sensors monitoring, data filtering, thresholds matching, synchronization.

Our prototype is, however far from being a complete solution. Some issues need to be handled, including the correct filtering of messages by the P4, enable real time communication with the robot (different approach than or steps solution for movement), it is necessary to explore the URSim nested package structure to use in the P4, and so on. Indeed, more challenges will arise while increasing the use cases and the operations to be performed by the P4. We expect to continue contributing to the evolution of the application's control by the network. All in all, our P4-based robotic control experiments bring to light new opportunities and challenges, not limited to robots, conversely, with a wide range of applications (e.g. industrial automation, real-time control) where low and bounded latency are paramount.

ACKNOWLEDGMENTS

This work was supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil under grant agreement UNI.63.

REFERENCES

[1] I. Parvez et al. 2018. A survey on low latency towards 5G: RAN, core network and caching solutions. *IEEE Communications Surveys & Tutorials* 20, 4 (2018).

[2] M. Agiwal et al. 2016. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys Tutorials* 18, 3 (thirdquarter 2016).

[3] J. T. Hagen and B. E. Mullins. 2013. TCP veto: A novel network attack and its Application to SCADA protocols. In *2013 IEEE PES ISGT Conference*. 1–6.

[4] P. Bosshart et al. 2014. P4: Programming Protocol-independent Packet Processors. *SIGCOMM* (July 2014), 9. https://doi.org/10.1145/2656877.2656890

[5] Universal Robots. 2015. Universal Robots Support. https://www.universal-robots.com/. https://www.universal-robots.com/