ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Technical Report

(03/2026)

QSTR-AL-PRC

Application layer protocol for robotics control

Technical Report ITU-T QSTR-AL-PRC

Application layer protocol for robotics control

Summary

With the rise of telepresence services, remote robotics control has emerged as a critical challenge in telecommunications.

Technical Report ITU-T QSTR-AL-PRC describes the application layer protocol for robotics control (PRC), designed for managing individual robots or clusters over International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) and transmission control protocol / Internet protocol (TCP/IP) networks. A key feature of the PRC is its optimization algorithm, which selects the optimal size of command series based on execution error probabilities to maximize transmission efficiency. The protocol addresses critical issues such as hardware calibration errors and software bugs by implementing a robust sequence of command confirmation and execution re-requests. Designed for Industry 4.0 and industrial Internet of things (IIoT) environments, the PRC provides a standardized framework for multi-vendor robotic interoperability while ensuring high reliability and deterministic performance.

Keywords

PRC, remote control, robotics, robots, telecommunication, telepresence services.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Table of Contents

Technical Report ITU-T QSTR-AL-PRC

Application layer protocol for robotics control

1 Scope

This Technical Report describes the application layer protocol for robotics control (PRC) as an application layer protocol for control of robots or their communities over a network remotely.

The Technical Report provides a description of the case when the protocol implements an algorithm for selecting a series of commands of optimal size that can be sent to the robot for execution, and also implements confirmation of the receipt of a series of commands and the execution of a series of commands.

The following are covered by the Technical Report:

– gap analyses on the issue;
– diagram of robot control over the network;
– protocol placement in the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) protocol stack;
– data transmission sequence.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following terms defined elsewhere:

3.1.1 industrial Internet of things (IIoT) [b-ITU-T Y.4003]: An Internet of things based enabling approach for industrial transformation, by taking advantage of existing and emerging information and communication technologies.

NOTE 1 – Emerging information and communication technologies include technologies for smart machines, robots, advanced industrial networks, industrial cloud computing and industrial data processing.

NOTE 2 – The industrial transformation enabled by the industrial Internet of things empowers the industry with, but not limited to, improved efficiency, intelligent production, reduced energy consumption, advanced collaboration modes and new business models. Industrial Internet of things enables smart manufacturing providing enhanced capabilities in support of manufacturing.

3.1.2 industrial Internet [b-ITU-T Y.2623]: A kind of industrial application ecology, characterized by convergence of the Internet and the new generation information and communication technologies (ICTs) with industrial systems, which serves as the key comprehensive information infrastructure for industrial intelligent development.

3.1.3 robot [b-ISO 8373]: Programmed actuated mechanism with a degree of autonomy to perform locomotion, manipulation or positioning.

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

IEC International Electrotechnical Commission

IIoT Industrial Internet of Things

ISO International Organization for Standardization

MTBF Mean Time Between Failures

PRC application layer Protocol for Robotics Control

QUIC Quick UDP Internet Connections

TCP Transmission Control Protocol

UDP User Datagram Protocol

5 Overview

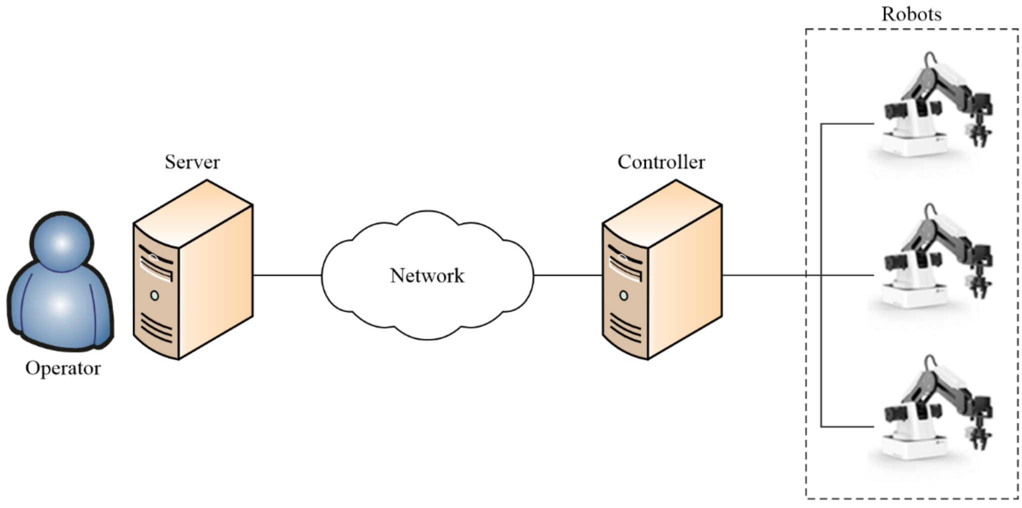
Today, due to the more intensive application of robots and their communities in the fourth industrial revolution and the transition to the practical realization of 2030 networks (with IIoT cloudification) [b-FG-NET2030-Sub-G1], it is an urgent task to develop network protocols that are designed to remotely control robots over a communication network. Since such robots can be used in industry, agriculture, healthcare, logistics, etc., it is necessary to develop such a protocol that can deliver control commands from the server to the robots as efficiently as possible. It should be kept in mind that when controlling a robot over a network, accuracy, correctness and timeliness of command execution are important. An important factor to be considered is the error of execution of a command or series of commands by the robot.

This can be due to poor calibration of the robot, software bugs, worn mechanisms and so on. In the case of an individual robot or cluster, errors in the execution of incoming control commands result in the need to reboot the robot or cluster and resend a series of commands to it.

6 Robot control over the network

6.1 Diagram of robot's control

Figure 1 shows a diagram which describe the robot's control over a network. Control commands are received from the server to the controller. In this case, the operator performs the overall system management and monitors the execution of the robot control commands. The controller is provided with a network interface as well as interfaces for connecting to the robots. One controller controls one robot or a community of robots, which thus form clusters.



QSTR-AL-PRC(26)

Figure 1 – Diagram of robot control over the network

6.2 PRC protocol message formats

6.2.1 General structure of the packet

Figure 2 shows the PRC protocol message formats.

Packet type	Session ID	Variable part
3 bits	8 bits	N bits

QSTR-AL-PRC(26)

Figure 2 – General header structure

Below is a description of the relevant fields.

The "packet type" field, 3 bits long, defines the type of packet being transmitted. Accordingly, the current version of the protocol can define 8 types of packets (messages), of which 4 are currently used.

"Session ID" is required while the session between the robot and controller is created. The 8-bit "session ID" field defines the communication session between the server and a specific robot connected to the controller. Accordingly, up to 256 robots can be connected to one controller in the current version of the protocol, which is sufficient given that there are no restrictions on the number of controllers. The length of cables or a section of the radio channel when the robots connecting to the controller will have natural limitations caused by signal attenuation and impact of natural and man-made interference.

6.2.2 Connection setup packet

Connection setup packet transmitted over the network from the server to the controller in order to establish a connection and agree on the main parameters of the PRC protocol.

Packet type	Session ID	Parameters
0 × 01	8 bits	N bits

QSTR-AL-PRC(26)

Figure 3 – Connection setup packet

At the moment, the following parameters can be used when transmitting this packet, which are sequentially recorded in the "parameters" field:

- SDC – the size of one control command, which depends on the robot application used,
- L – the total size of service information in control commands transmitted within one series of commands,
- Per – the probability of an error in executing a single command by the robot.

The length of the "parameters" field is not limited, which allows you to add additional parameters when establishing a communication session between the server and the controller, if necessary. The structure of the "parameters" field is an alternation of "parameter"/"value" sub-fields, each 16 bits long, for recording the necessary parameters and their values.

6.2.3 Data packet

Data packet transmitted over the network from the server to the controller and contains a series of control commands of the established length.

Packet type	Session ID	Packet number	Number of commands	Data
0 × 02	8 bits	24 bits	10 bits	N bits

QSTR-AL-PRC(26)

Figure 4 – Data packet

This packet contains a "packet number" field of 24 bits, which is used to number the data packets sent by the server, which allows (taking into account the possibility of cyclic numbering of transmitted packets) to avoid generating packets with the same numbers during one session.

This packet also contains a 10-bit "number of commands" field, which determines the number of control commands sequentially written in the data field. Accordingly, in the current version of the protocol, no more than 1024 commands can be transmitted in one packet, which is a reasonable limitation on the maximum size of the PRC application layer protocol data packet.

The "data" field is intended for sequential recording of control commands. The size of the field is not directly limited. However, the size of the "data" field is indirectly limited by the size of one command and the maximum number of commands in a series.

6.2.4 Confirmation packet

Confirmation packet transmitted from the controller to the server and depending on the set flags, can confirm the receipt of the main parameters, as well as the reception or execution of a series of commands.

Packet type	Session ID	Confirmation type	Confirmation number	Confirmed command number
0 × 03	8 bits	3 bits	24 bits	10 bits

QSTR-AL-PRC(26)

Figure 5 – Confirmation packet

The 3-bit "confirmation type" field allows you to specify up to 8 types of service messages, including types of confirmation messages transmitted from the controller to the server. Currently, 3 types of confirmation packets are defined: confirmation of receipt of a connection setup packet with basic parameters, confirmation of receipt by the controller of a data packet with a series of commands, confirmation by the controller of error-free execution of a series of commands by the robot.

The 24-bit "confirmation number" field is used to confirm receipt of a data packet with a series of control commands. Upon receipt of a packet with number N from the server, the controller responds with a packet with number N.

The 10-bit "confirmed command number" field is used to confirm execution of a control command with the specified number. This is necessary in the case when an error in executing a command by the robot occurs after executing one of the commands in the series.

6.2.5 Connection check packet

Connection check packet transmitted to check the state of the network between the server and the controller, as well as to check the operability of the controller itself or the robots connected to it.

Packet type	Session ID	Flags
0 × 04	8 bits	8 bits

QSTR-AL-PRC(26)

Figure 6 – Connection check packet

The "flags" field of the connection check packet is 8 bits long and is used to select the appropriate options. Having received such a message, the controller understands what operation the server is requesting based on the presence of "1" values in the corresponding bits of the "flags" field. Examples of flags:

- 00000001 – connection check between the controller and the server;
- 00000010 – connection check between the controller and the robots;
- 00000100 – connection check between the controller and the time synchronization server.

The remaining values are reserved for possible future uses.

7 PRC placement in ISO/IEC stack

The PRC protocol is designed to operate at the application level using the ISO/IEC and TCP/IP models. At the transport level, well-known standard network protocols UDP, TCP, quick UDP Internet connections (QUIC) can be used to transmit data.

At the same time, user datagram protocol (UDP) can be used to transmit PRC protocol data packets in small networks with stable delay, low jitter, and a minimum level of data packet loss.

The TCP transport protocol can be used to transmit PRC protocol data packets in medium and large corporate networks with average values of delay and jitter, etc.

The QUIC protocol is supposed to replace the UDP, TCP protocols in future networks, but additional research is needed.

8 Data transmission sequence

8.1 General description

Robotic integration into the communication network is facilitated by a controller. Controller bridges specialized industrial robot interfaces with standard enterprise network interfaces (e.g., [b-IEEE 802.3]). Support for a broad spectrum of industrial protocols is essential for multi-vendor robotic interoperability. Furthermore, the open PRC protocol should be implemented at the controller's software level to streamline integration with established industrial network infrastructures.

Figure 7 shows the PRC protocol data transmission sequence. A series of control commands are transmitted from the server to the controller. The controller then acknowledges receipt of the series of commands. After the series of commands is executed, the controller confirms the execution of the series of commands by the robot by sending a confirmation message.

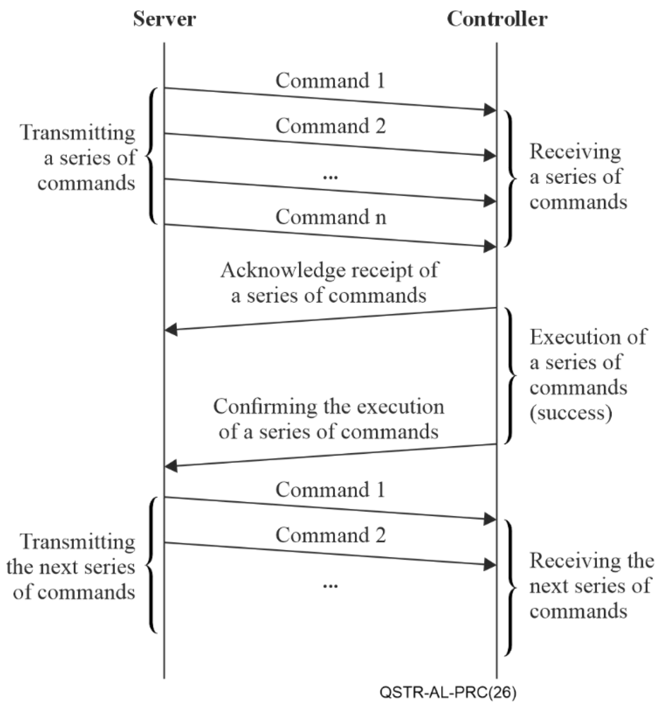


Figure 7 – Data transmission sequence (successful execution of a series of commands by the robot)

In Figure 8 shows a sequence for a case where an error occurs while the robot is executing a series of commands, followed by a reboot of the robot and a re-request of the series of commands.

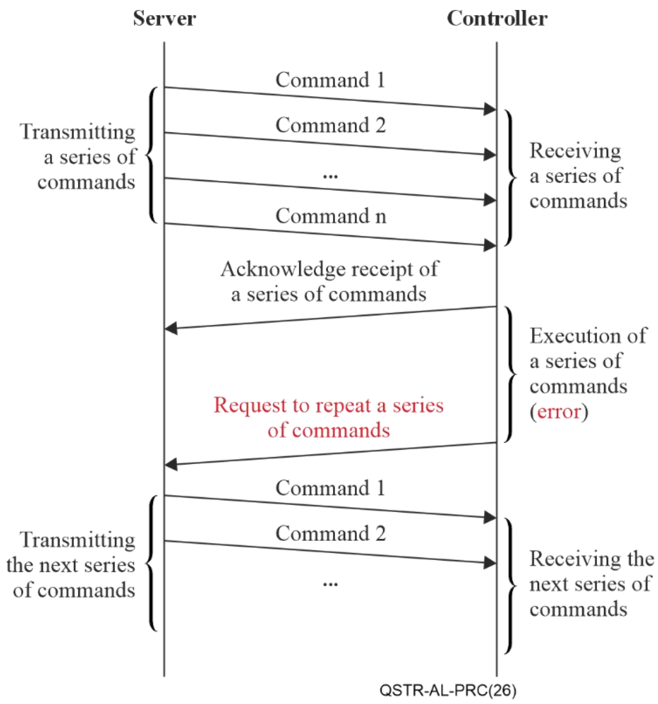


Figure 8 – Data transmission sequence (error of the robot executing a series of commands)

8.2 Implementation features

The commands, which are shown on the diagram (Figure 8) should not be uniquely associated with individual messages or data packets, which may also be a special case. In general, commands can be transmitted either in separate packets or several commands in one packet. This depends on the characteristics of the commands and the design of the control system.

Controller receives a sequence (series) of commands from the control system and places them in a buffer. Then the controller informs the server that a series of commands has been received and placed in the buffer. After this, the controller executes commands, sequentially selecting them from the buffer. When the entire sequence is completed, controller sends a corresponding message to the server about its readiness to accept the next sequence of commands.

If an unsuccessful event occurs during the execution of commands, that is, the robot under the control of the controller is unable to execute any i-th command in the sequence, then the contents of the buffer are cleared, and the controller sends a message to the server about the need to repeat the sequence of commands.

From the presented model it is clear that its functioning depends on the probability of an unsuccessful execution of the command. The efficiency as the share of useful data in the total volume of transmitted data:

$$e = \frac{U}{A} \tag{1}$$

where U – the volume of useful data transferred over the network; A – total data volume.

Payload refers to commands sent, received, and successfully executed in the system. A proposed algorithm for selecting the command series size is presented below.

9 Selection of command series size

The efficiency dependence is determined by formula (1) and depends on the probability of robot error when executing a command and the size of the transmitted data. The payload size can be defined as the total amount of data for all executed commands:

$$U = \sum_{j=1}^{m} S_{DC_j} \text{ , bytes} \tag{2}$$

where S_{DC_j} – is the data volume of the *j*-th command (bytes); m – number of successfully executed commands.

With the same size of all commands $S_{DC_j} = S_{DC_i}, i, j = 1, \ldots, m$

$$U = mS_{DC}, \text{ bytes} \tag{3}$$

All commands are transmitted sequentially (as a series) of *b* commands, where *b* is the number of commands placed in the controller buffer. Next, we shall work with a series of commands, equating *m* in formula (2) to *b: m = b*. Then, taking into account condition (3):

$$U_S = bS_{DC} \text{ bytes,} \tag{4}$$

where *b* is the number of commands in the series.

The calculation of the total data volume is based on the assumption that there is a non-zero probability of error (failure to execute) of a command by the robot, and this probability is the same for all commands. It is assumed that the error probabilities are independent. If an execution error occurs, the entire command sequence is removed from the controller's buffer and re-requested and transmitted from the server to the controller. Thus, in the general case, successfully executing a command sequence may require k attempts (transmissions). Then the total data volume in (1) will be determined as:

$$A = (U_s + L)\bar{k} \text{ , bytes} \tag{5}$$

where L is the amount of data in service messages (requests and confirmations) per one series of commands; $\bar{k}$ the average number of passes of a series of command. The mean is:

$$\bar{k} = \frac{1}{(1-p_{er})^b} \tag{6}$$

where p_{er} is the probability of an error in executing the command by the robot; $(1 - p_{er})^b$– the probability that a series of commands will be executed. At the same time, for practice scenarios $0 < (1 - p_{er})^b < 1$. For industrial robots in the mid-price segment, the mean time between failures (MTBF) typically ranges from approximately 10 000 to 50 000 hours [b-Kampa]. In the context of Industry 4.0, the increasing deployment and large-scale adoption of robotic systems are expected to lead to a reduction in their cost, which may be accompanied by a higher probability of failures or operational errors [b-Asari]. Furthermore, it should be taken into account that when multiple robots sequentially perform operations within a manufacturing process (e.g., during parts production), the probability of failure or error in the overall processing chain increases.

For the general case, it should be noted that if the probabilities of failure to execute commands are different, then (6) will have the form:

$$\bar{k} = \frac{1}{\prod_{i=1}^{b}\left(1-p_i^{(er)}\right)} \tag{7}$$

where is $p_i^{(er)}$the probability of error when executing the *i*-th command from the series. Then the channel efficiency, defined as (1) taking into account (3) – (7), can be calculated as

$$e(b) = \frac{bS_{DC}\prod_{i=1}^{b}\left(1-p_i^{(er)}\right)}{bS_{DC}+L} \tag{8}$$

Or for the case of equal error probabilities $p_{er} = p_i^{(er)};\ i = 1, \ldots, m$

$$e(b) = \frac{bS_{DC}(1-p_{er})^b}{bS_{DC}+L} \tag{9}$$

From formulas (8) and (9) it is clear that if the error probability is zero, then the efficiency depends only on the share of the overhead data and tends to one as the packet size increases. From these expressions, it can be noted that with a non-zero error probability, the efficiency of using the communication channel changes when the size of a series of commands changes, and this dependence has a maximum value that can be found as a solution to the expression:

$$b^* = \arg\max_b e(b); \quad b \ge 0 \tag{10}$$

Solution (10) can be obtained by the classical method of finding the extremum of a function and results in an expression for b^*:

$$b^* = -\frac{L}{2S_{DC}} - \frac{\sqrt{L^2 \ln(1-p_{er})^2 - 4LS_{DC}\ln(1-p_{er})}}{2S_{DC}\ln(1-p_{er})} \tag{11}$$

Based on formula (11) it is clear that the optimal number of commands in the series b^* depends on the probability of a robot error, the amount of transmitted data (command series sizes S_{DC}) and the amount of service data L. The amount of service data is determined by the size of the confirmation or re-request messages.

From the above dependence it is clear that the optimal number of commands in a series decreases with an increase in the probability of error when executing a command by the robot and with an increase in the volume of data in the command. The efficiency gain achieved by this method can be estimated as:

$$\Delta e = \frac{e(p_{er},b^*) - e(p_{er},1)}{e(p_{er},1)} 100\% \tag{12}$$

where $e(\ldots)$ – is determined by formula (9).

The point of comparison in this case is the mode when $b = 1$.

During the operation of a robotic system, changes in the probability of errors in performing operations may occur, for example, due to changes in external influencing factors. Therefore, during its operation, it is advisable to keep error statistics and adjust the values of the corresponding probabilities.

We will assume that the probability of error is determined by the model of independent trials and can be described by the Bernoulli scheme. Then the confidence interval for probability will be determined as (formula 13):

$$\bar{p}_{er} = \tilde{p}_{er} \pm G_{1-\alpha/2}\sqrt{\frac{\tilde{p}_{er}(1-\tilde{p}_{er})}{n}} \tag{13}$$

where $\tilde{p}_{er} = \frac{n_{er}}{n}$, n – total number of observations (operation executions); n_{er} – number of errors when performing the operation; $G_{1-q/2}$ – quantile of normal distribution for confidence probability (1–α/2).

Then the reliability of the estimate according to formula (12) can be described by the relative error of the probability estimate as:

$$\delta = 100\, G_{1-a/2}\sqrt{\frac{1-\tilde{p}_{er}}{n\tilde{p}_{er}}}\ \% \tag{14}$$

It is advisable to collect statistics cyclically and update the error probability values when the reliability of its statistical assessment is sufficient, i.e., the value determined by (13) does not exceed the specified permissible error $\delta \le \delta_0$.

The algorithm is shown in Figure 9.

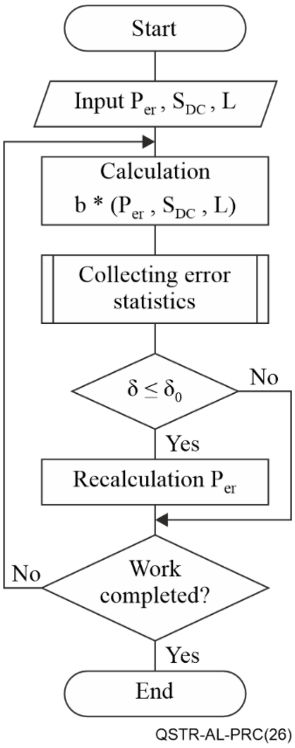


Figure 9 – The algorithm

The message exchange scenario for the proposed algorithm is shown in Figure 10.

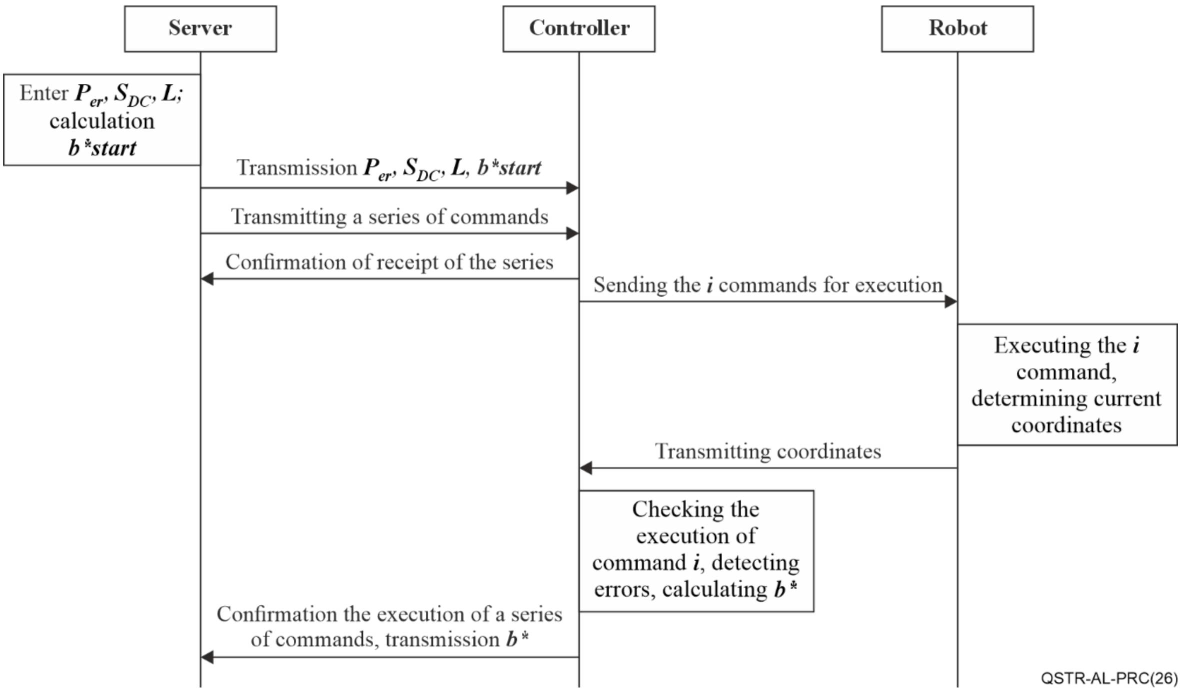


Figure 10 – Message exchange scenario for the proposed algorithm

Appendix I

Use case

Industry 4.0 is seeing the widespread adoption of robotics to replace manual labour. In modern industrial Internet of things (IIoT) networks, locally deployed identical robots performing similar tasks form stable clusters. In this case, robot clusters are managed by controllers that, on the one hand, connect the robots to the network using the required network interface, and, on the other, provide centralized cluster management and command transmission between the server and the robot cluster.

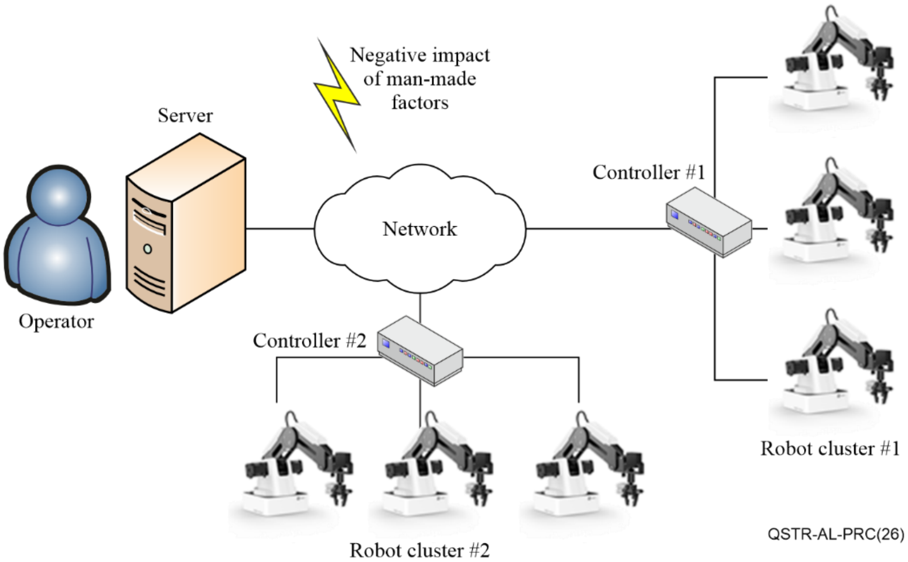


Figure I.1 – Use case of network robot clusters control

However, different robots in the same factory have different characteristics (e.g., available memory capacity for storing incoming commands) and different command execution times. Furthermore, robot clusters in the same factory may be exposed to different environmental conditions, such as constant or intermittent exposure to electromagnetic interference, climatic conditions, etc. These factors can impact both the robots themselves and the communication channels and the telecommunications equipment.

Accordingly, the PRC protocol enables centralized management of robot clusters and flexible adjustment of data transmission parameters (e.g., command series size) to avoid network congestion or to adapt to increased packet loss.

Appendix II

Gap analysis

One of the main goals of Q2/11 is to develop signalling requirements and protocols to support future services in telecommunication networks evolving to IMT-2020 network and beyond. Taking into account the deep integration of IoT in different areas it is necessary to consider the industry part. IIoT with cloudification is one of the major directions in network 2030 [b-FG-NET2030-Sub-G1]. Thus, the effective protocols for remote robot(-s) control is a very important and relevant task in the field of smart manufacturing [b-ITU-T Y.4003].

Regarding the IIoT with cloudification issues, there are follow key network requirements defined in the Technical Report from Focus Group "Network 2030" (2020 [b-FG-NET2030-Sub-G1]):

– Latency. IIoT systems contain many control systems and subsystems, including the high-level risks industry-secure systems. Here the cycle times ranging from sub-ms to 10 ms.
– Synchronization. It is necessary to provide cooperation between IIoT devices, systems, etc. In order to ensure the correct implementation of production processes, robotics systems and management clouds must have a high level of synchronisation, including some risks from internal and external fluences.
– Large-scale deterministic networking capability. Due to the cloudification in industrial IIoT and related systems, the aforementioned network requirements should be supported by large-scale deterministic networks in the network 2030.
– Security and reliability with 99,999999% availability requirement for IIoT applications. And also small and bounded jitters.

There are high level requirements presented above, refer to the ITU-T technical report, which describes the future use cases of 2030 networks.

At the current time, we found a few recommendations, related to the proposal with a new protocol for remote robotics control, which can also be used in the IIoT cloudification concept. The gap analysis is presented in the following table.

Table II.1 – Gap analysis: Related works for application layer protocol for robotics control

No	SDOs	Title	Status	Analysis results
1	ITU-T SG20	Supplement 63 to ITU-T Y-series Recommendations Unlocking Internet of things with artificial intelligence [b-ITU-T Y Suppl. 63]	Agreed, In force	This Recommendation examines the question of how artificial intelligence could be used to implement the Internet of things (IoT) in order to transition to "smart cities". The main elements discussed in this supplement are: various technological implementations of artificial intelligence (AI) that can contribute to the transformation of smart cities; the role played by AI in managing data generated in the Internet of things and urban spaces; the main advantages of the introduction of artificial intelligence and the study of how this technology could be used to achieve the sustainable development goals (SDGs). As a result of the analysis of this Recommendation, it can be concluded that robotics is considered by the authors as one of the

Table II.1 – Gap analysis: Related works for application layer protocol for robotics control

No	SDOs	Title	Status	Analysis results
				key directions of the Internet of Things in the context of smart cities and smart industries, as well as one of the most important areas of practical application of artificial intelligence technologies. The presented draft recommendation describes a model network for complex testing of various robotics in the case of remote control over the network.
2	ITU-T SG2	Recommendation [b-ITU-T M.3367] Requirements for robot-based on-site smart patrol of telecommunication networks	Agreed, In force	This Recommendation describes how intelligent maintenance robots (IMRs) can be used for intelligent patrolling of telecommunication networks. In doing so, this Recommendation describes that IMR-based intelligent patrolling of telecommunication networks includes network elements to be patrolled, requirements for an IMR-based patrolling control function and corresponding control patrols. As a result of the analysis of this Recommendation, it can be concluded that IMR-based smart patrol system should operate in an autonomous mode, with the routing and recognition of objects occurring automatically on the basis of special algorithms. Accordingly, it can be concluded that the issues of remote control of IMR-based smart patrol system elements (including robots themselves, as well as infrastructure elements) become very relevant for further study and development.
3	ITU-T SG13	Supplement 72 to ITU-T Y-series Recommendations ITU-T Y.3000-series – Artificial intelligence standardisation roadmap [b-ITU-T Y Suppl. 72]	Agreed, In force	This Supplement provides a roadmap for standardisation for artificial intelligence (AI) in information and communication technology. This AI standardization roadmap has been developed to assist in the development of AI-related standards by providing information on existing standards and the development of key standards. Many of the papers reviewed in this roadmap describe different aspects of the application of current AI technologies to robotics: natural language processing recognition, computer vision, and machine learning. Accordingly, it can be concluded that one of the foundations for future large-scale robotics deployment will be ubiquitous AI technologies.

Table II.1 – Gap analysis: Related works for application layer protocol for robotics control

No	SDOs	Title	Status	Analysis results
4	ITU-T SG13	Recommendation [b-ITU-T Y.2623] Requirements and framework of industrial Internet networking based on future packet based network evolution	Agreed, In force	This Recommendation provides definitions and requirements for industrial Internet networks to support customised, collaborative, service-oriented and intelligent manufacturing/services. The text also describes the structure of the industrial Internet to understand the essential relationships between entities on the internal enterprise network and the external enterprise network. This Recommendation also introduces several new terms such as industrial Internet, intelligent manufacturing, factory internal and external networks, which is very important for their further use in the development of technical standards. After a careful study of this standard, which in this list also describes network interaction with industrial robots, it was found that in application level protocols the authors consider well-known and widespread protocols such as HTTP and MQTT, which, despite all their advantages do not have specialised network protocols for controlling robots within the framework of the industrial Internet of things (IIoT). Therefore they do not take into account the peculiarities of the shape of the robot itself (failures and malfunctions in its operation), the peculiarities of connecting robots or their clusters to the network (via heterogeneous gateways), as well as the peculiarities of transmission over the network robot control commands.

Bibliography

[b-ITU-T M.3367] Recommendation ITU-T M.3367 (2023), *Requirements for robot-based on-site smart patrol of telecommunication networks.*

[b-ITU-T Y.2623] Recommendation ITU-T Y.2623 (2021), *Requirements and framework of industrial Internet networking based on future packet based network evolution.*

[b-ITU-T Y.4003] Recommendation ITU-T Y.4003 (2018), *Overview of smart manufacturing in the context of the industrial Internet of things.*

[b-ITU-T Y Suppl. 63] Supplement 63 to ITU-T Y-series Recommendations (2020), *Unlocking Internet of things with artificial intelligence*.

[b-ITU-T Y Suppl. 72] Supplement 72 to ITU-T Y-series Recommendations (2020), *Artificial intelligence standardization roadmap*.

[b-FG-NET2030-Sub-G1] ITU-T Technical Report FG-NET2030-Sub-G1 (2020), *Representative use cases and key network requirements for Network 2030.*
<https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-NET2030-2020-SUB.G1-PDF-E.pdf>

[b-IEEE 802.3] IEEE 802.3-2022, IEEE Standard for Ethernet.
<https://standards.ieee.org/ieee/802.3/10422/>

[b-ISO 8373] International Standard ISO 8373:2021, *Robotics – Vocabulary.*
<https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en>

[b-Asari] Asari, R. (2018), *Automotive Industrial Robot – Total Cost of Ownership (TCO).*
<https://www.researchgate.net/publication/327112724_Automotive_Industrial_Robot_-_Total_Cost_of_Ownership>

[b-Kampa] Kampa, A. (2018), *The Review of Reliability Factors Related to Industrial Robots. Robotics & Automation Engineering Journal.*
<https://juniperpublishers.com/raej/pdf/RAEJ.MS.ID.555624.pdf>