Bejarano, Ronal; Pääkkönen, Roope; Olaf Blech, Jan; Peake, Ian; Herrmann, Peter; Vyatkin, Valeriy

Assessing Long Distance Communication Alternatives for the Remote Control of AGVs

Published in:
Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2020

DOI:
10.1109/ETFA46521.2020.9211954

Published: 01/09/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Bejarano, R., Pääkkönen, R., Olaf Blech, J., Peake, I., Herrmann, P., & Vyatkin, V. (2020). Assessing Long Distance Communication Alternatives for the Remote Control of AGVs. In *Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2020* (pp. 69-76). Article 9211954 (Proceedings IEEE International Conference on Emerging Technologies and Factory Automation). IEEE. https://doi.org/10.1109/ETFA46521.2020.9211954

Assessing Long Distance Communication Alternatives for the Remote Control of AGVs

Ronal Bejarano*, Roope Pääkkönen*, Jan Olaf Blech*, Ian Peake‡, Peter Herrmann§, Valeriy Vyatkin*,**
* School of Electrical Engineering, Aalto University, Espoo, Finland
Email: {ronal.bejarano, roope.paakkonen, jan.blech, valeriy.vyatkin}@aalto.fi
‡ RMIT University, Melbourne, Australia
Email: ian.peake@rmit.edu.au
§ Norwegian University of Science and Technology (NTNU), Trondheim, Norway
Email: herrmann@ntnu.no
** Luleå University of Technology, Luleå, Sweden

***Abstract*—Remote monitoring and control of factory equipment promises a more streamlined and therefore less expensive system operation and maintenance. The geographical distance between a factory and its control center, however, may influence the Quality of Service parameters of the network connections which might stymie the overall control process. To get a better understanding of these potential issues and their impact, we conducted a series of measurements over varying distances for the remote control, operation and simulation of Automated Guided Vehicles (AGVs) that are often used in modern factory environments. To achieve these tests, we defined three communication patterns reflecting local and remote connections as well as the usage of cloud-based services. Applying these patterns, we connected the Factory of the Future at the Aalto University in Finland with the VxLab at the RMIT University in Australia and the Microsoft Azure cloud in the Netherlands. This allowed us to measure important Quality of Service networking parameters for the communication over short, medium, and very long distances. In this paper, we present first empirical results and discuss their impact on the remote control of AGVs.**

***Index Terms*—Automated Guided Vehicles (AGVs), communication patterns, cloud computing, remote control, operation and maintenance, virtual private networks (VPNs)**

I. Introduction

Remote operation, maintenance and, to some extent, commissioning of factory or mining equipment has aroused interest in the last decade. This holds particularly for the manufacturing, material handling, and mining industries (see, e.g., [1]). The remote operation is supported by rising levels of digitization, in particular by trends such as Industry 4.0 and the Industrial Internet of Things (IoT). Also, recent events such as the COVID-19 pandemic increase the attractiveness of such technologies since they may help to reduce travelling of operation personal as well as close encounters of humans.

Nevertheless, to be viable, remote maintenance and operation has to meet some important conditions. For instance, a potentially large number of autonomous entities such as AGVs and mobile robots may be used on a plant site, often in the vicinity of humans that have to be protected. Thus, the overall system must guarantee that human operators and technicians can react on incidents under very hard real time conditions. In addition, a remote control system has to be able to be embedded in existing infrastructures that are operated using legacy application architectures and protocols. Also the data to be transferred can be rich, e.g., when video streams are used to enable the human operator to monitor the operation process visually.

All these aspects call for a good and reliable performance of the underlying networks. An aggravation is the heterogeneity of many networks that are often assembled from a variety of wide-area and local-area computer networks (such as 5G, WiFi, and field buses). Moreover, various cloud and edge data processing techniques might be involved as well.

An often useful way to meet these challenges is to replicate the situation locally on the operator's control center in order to anticipate potentially critical developments in due time. Such technologies as digital twins and online simulation [2], synchronised with the physical process, can be seen as elements helping to find a solution.

The core of any solution, however, is to be sure that the networks connecting a plant with a control center fulfill certain dedicated Quality of Service (QoS) parameters. Only then, a timely reaction on critical developments is guaranteed. To address the complexity issues of the computer networks mentioned above, we introduce a set of communication patterns in this paper. They cover some typical network layouts for remote control. In particular, both local and remote connections as well as the use of cloud services are reflected.

The main contribution of the paper is the presentation of a number of initial tests, in which we gathered QoS parameters across three communication and computing infrastructures. We see the results of these tests as a starting point to investigate the entire remote failure resolution scenario for a set of AGVs. The equipment used for our tests resides on two sites, the Factory of the Future [3] at the Aalto University in Espoo (Helsinki), Finland, and the Virtual Experiences Laboratory (VXLab) [4] at RMIT University in Melbourne, Australia. Furthermore, the Microsoft Azure cloud with its hosting site in Amsterdam, the Netherlands, is used. We present empirical results on latency, throughput, and availability that shall serve as a guide for other sites and for our own future extensions.

The article is structured as follows: After sketching some related work in Sect. II, we introduce the communication patterns and the equipment used in Sect. III. Thereafter, the description of the conducted tests is presented in Sect. IV while, in Sect. V, we describe the evaluation criteria used in our tests. In Sect. VI, the results of the tests and their impact on the use of remote control technologies is discussed followed by some concluding remarks in Sect. VII.

II. Related Work

Over the last decade, several authors approached the challenges of controlling mobile robots remotely in multiple ways. The often autonomous operation of a robot requires special techniques to connect it with the external world, in particular, the use of wireless communication channels. This situation fostered the adoption of standard ICT technologies into commercial AGV models (e.g., Wi-Fi, Service Oriented Architecture, Web Services), which also improves the compatibility with standard computing systems. However, the use of Wi-Fi networks can produce a negative impact, especially when the controllers are not on-board. As discussed in [5], delay correction techniques to mitigate the impact of the variable delay over the Wi-Fi control network of typically 5 to 95 ms, are required. The presented techniques include a network delay estimator and a position predictor, to effectively operate a controller with a period of 100 ms. In [6], a list of essential processes executed by AGVs is introduced. It includes task allocation, localization, path planing, motion planning, and vehicle management. The processes require to compute control algorithms and techniques that demand information exchange, and sometimes, even overcome the lack of it.

In practice, the use of a fleet of AGVs requires the integration of networked collaborative interfaces to support the autonomous operation of AGVs. As shown in [7], a centralized cloud service acting as a control center can bring benefits on efficiency and flexibility of the AGV motion coordination in a warehouse environment. The cloud denomination in this work is based on a private pool of computational resources accessible by a Wi-Fi network with average latency values of 43 milliseconds. It is suitable for systems with update rates of 10 Hertz presenting a similar situation as the studies aforementioned, in which the use of networked features puts an undesirable delay on the system.

In previous work [8], we connected the visualization infrastructure of VXLab (see Sect. III) with a Lego Mindstorms-based train system that was positioned in Trondheim, Norway [9]. This connection was based on the server-based IoT protocol AMQP [10] and allowed the remote monitoring and, with some time-based limitations, controlling of the trains from Melbourne. We installed a remote AMQP server in the Australian cloud infrastructure Nectar, which was connected to both the VXLab and the controllers and sensors used in the trains. Status information like the current position and speed of a train on a track or the settings of the switches, was directly sent to the remote AMQP server, which made direct monitoring from the VXLab possible. Likewise, control commands issued at the VXLab were sent via the remote server to the controllers on the train system. To get an idea about the transmission delays, intensive round trip time latency tests were carried out, see also [11]. Figure 1 depicts a 24 hour test between the lab in Trondheim and the remote AMQP server in the Nectar cloud. Here, a ping message was sent every two seconds for a whole day to get an idea if the round trip time is fluctuating. The figure reveals that the transmission delay was quite stable between 350 and 360 milliseconds. We experienced only very few fluctuations that never exceeded 885 milliseconds. The facilities of the VxLab were also utilized in [12]. Here, we investigated a software architecture for controlling robots that take advantage of the equipment offered in the VxLab.

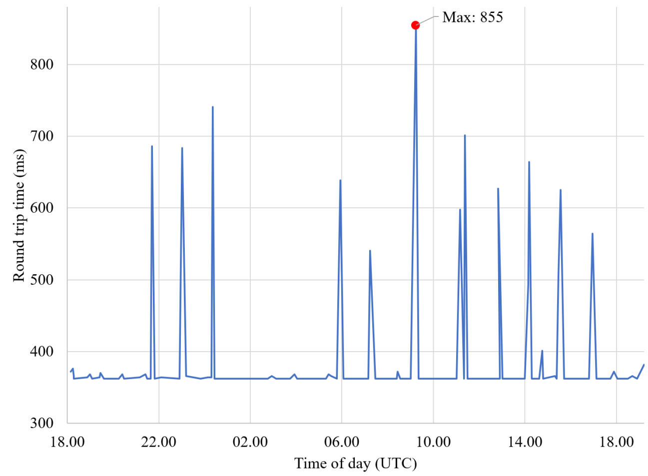

Fig. 1. Tests NTNU-RMIT, see [8]

III. Communication architecture

Our tests follow three main communication patterns that are tailored to the use in IoT and Industry 4.0 contexts. They are depicted in Fig. 2 and introduced in the following:

The access to many IoT devices is nowadays provided by cloud-based services, see, e.g., [13]. This is reflected by the *cloud communication pattern* on the left side of the figure. The sensor readings and actuator commands of an AGV can be remotely accessed by devices using cloud data centers. The communication between these centers and the AGV is provided by a Virtual Private Network (VPN) that runs on top of the Internet.

Remote access on AGVs can also be carried out by connecting them directly with the control devices. This is addressed by the *remote communication pattern* shown in the center of Fig. 2. Here, the network is also realized by VPNs running on the Internet. The main difference is that the AGV is directly linked with a control unit and not a cloud service. In the last years, a number of specialized communication protocols for IoT and Industry 4.0 applications such as MQTT [14] and AMQP [10] emerged. ROS uses *topics* for this purpose. These mostly server-based protocols vastly facilitate the use of the

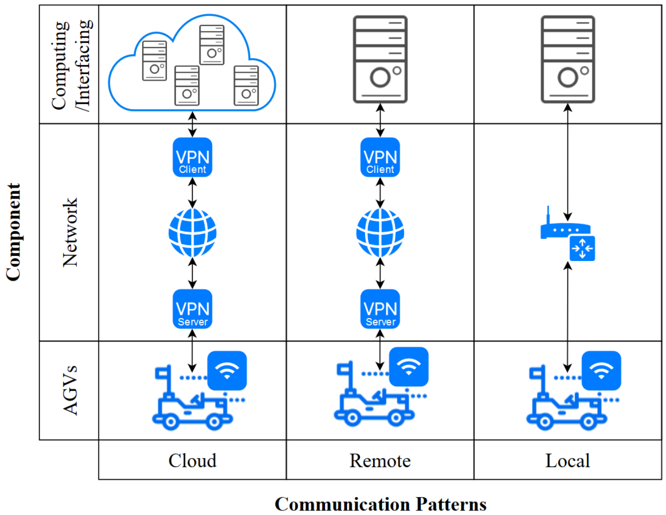

Fig. 2. Component mapping for communication patterns of remotely controlled AGVs

Fig. 3. Industrial AGVs at the Aalto Factory of the Future

remote communication pattern. An evaluation of the performance aspects of different protocols can be found in [15].

Of course, the access can also be local which is still the predominant way to operate AGVs. The *local communication pattern* at the right side of Fig. 2 can be used for this case. As a mobile device, the AGV is connected with control or supervision units via a Wireless Local Area Network (WLAN) either by a direct peer-to-peer connection or, as shown in the figure, using a bridge.

As mentioned in the introduction, we use the Aalto Factory of the Future, the VxLab, and the Microsoft Azure cloud to connect devices via the patterns introduced above. The Aalto Factory of the Future [3] is a facility for research, innovation and educational projects at Aalto University. In particular, software aspects of flexible, reconfigurable manufacturing scenarios are studied. Figure 3 shows two AGVs. The one at the left is a MIR 100 that can, e.g., be connected to laptops and Raspberry Pi single board computers.

The VXLab [4] is placed on the city campus of the RMIT University in Melbourne, Australia. It was developed to explore themes of software architecture and testing for remote monitoring and collaboration in the control automation industry. The lab consists of industry and collaborative robots, a seven meters long tiled display wall, and virtual reality equipment. These units are augmented by dedicated servers in the Cyber-physical Simulation Rack (CSRack) facility that consists of HP ProLiant blades in an RMIT data center. The CSRack hosts several projects including the Gazebo simulation engine (ROS Melodic, Ubuntu 18.04). Further, the VXLab has another MIR 100 as well as Rosie, the integration of a Baxter collaborative robot with a Dataspeed mobility base.

IV. Methodology and Test Setup

The software to test the network parameters was implemented in a Python 3 script that can be operated in the roles of both, a server and a client. A single test runs automatically every minute until it reaches the predefined number of iterations. At the end of each test run, the results are logged in a database file hosted by the client device, tagged with the UTC time of its execution.

We test the three relevant Quality of Service (QoS) parameters latency, throughput, and availability. To get the *latency* L, we executed the well-known networking utility ping with its default 56-bytes large packets 10 times, and compute the arithmetic mean of the response times. If $t_p(k)$ is the response time of the k'th ping run, the latency is calculated as follows:

$$L = \frac{1}{10} \sum_{k=1}^{10} t_p(k) \tag{1}$$

To calculate the *throughput* T of data via TCP connections, the points of time t_0, when a connection is initiated, and t_f, when its termination is confirmed, are extracted. When a TCP connection is established, we send the content of a buffer c times before ending the connection. Moreover, we experiment with different buffer sizes B that can be set to 1, 2, 4, 8, 16, 32 and 64 kilobytes. We can then determine the throughput using the following formula:

$$T = \frac{B * c}{t_f - t_0} \tag{2}$$

The TCP connection used in our tests was implemented by the Sockets module on Python.

The *availability* α of a connection corresponds to the ratio between successfully completed tests and all tests conducted. We consider a test as successful if the pings in the latency tests are confirmed and the throughput implementation does not fall below a certain threshold. Be s_u the number of successful and s_t the number of all tests carried out, the availability can be computed as follows:

$$\alpha = \frac{s_u}{s_t} \tag{3}$$

After completing the preselected testing iterations, the availability is computed from the results of the latency and throughput tests logged during the various test runs.

Our hypothesis is that using the existing AGV technologies with remotely located computing and interfacing resources leads to some subpar QoS parameter values. In particular, we fear that the higher complexity brought by the extended geographical reach of the data network and other external factors, not under control of the AGV user, may result in slower data transfer times and a reduced availability. That may jeopardize the proper execution of the networked features in AGVs. To find out if our assumptions are valid, we carry out the proposed tests that also allow us to evaluate and compare the influence of the communication patterns introduced in Sect. III.

To test our hypothesis on different geographical bases, we carry out three test scenarios that are highly different with respect to the distances between an AGV and its control devices. The first scenario is based on the remote communication pattern. It reflects a city environment, where the separation of the agents is less than 20 kilometers, but there is not a dedicated channel of communication. The second scenario incorporates a cloud environment following the cloud communication pattern. It connects devices at the Aalto Factory of the Future with one of the Microsoft Azure data centers in Amsterdam which are approximately 1,500 km apart from each other. The third scenario is used to test the connectivity over vast geographical distances. Here, we use the remote communication pattern to link Helsinki with computing resources located at the VXLab. The geographical separation between the two points is approximately 15,200 kilometers.

V. Evaluation Criteria

Carrying out the three scenarios will provide us with a better idea if the AGV realization using remote agents is feasible in practice. Moreover, we like to study the impacts of the following three important external influences:

Buffer size and total amount of data: TCP-based wide-area data traffic is usually realized by a sequence of data packets travelling to their recipient via a number of routers. To find out the effect of the amount of data sent in such transmissions via routers, it is worthwhile to vary the above discussed buffer sizes B and the number c of transmission iterations in a TCP connection. Our experiments refer to the work of Manzi et al. in [16] who reported about the amount of data required to stream velocity commands remotely for a mobile robot (DoRo). These Velocity commands have to be transmitted 64.5 times a second while the size of each message is 236 bytes. Thus, considering an overhead of 30%, the network connection has to guarantee the transmission of around 20 kilobytes per second. In our tests, we replicate this data transfer behavior by using a buffer size of 2 kB and transmitting its content 10 times in a single TCP connection.

Time of the day and network load: As proposed for the cloud and remote communication patterns (see Sect. III), we use VPN connections that run on external Internet connections.

TABLE I
Latency, throughput and availability results for the three test scenarios

Quality of Service parameter	Statistic	Test scenario		
		1. City	***2. Cloud***	***3. Remote***
Latency	μ (ms)	11.89	29.11	357.11
	max. (ms)	154.23	77.38	445.24
	min. (ms)	5.32	27.25	327.90
	σ	14.20	3.92	10.28
	c_v	119.5%	13.4%	2.8%
Throughput	μ (kB/s)	538.40	218.71	19.58
	max. (kB/s)	1324.70	239.82	20.53
	min. (kB/s)	1.28	18.64	2.30
	σ	173.33	20.38	0.73
	c_v	32.1%	9.3%	3.7%
Availability	Percentage	98.90%	99.93%	8.19%

An important factor for the connectivity parameters mentioned above is the performances provided by the Internet Service Providers (ISPs) which may heavily depend on the network load used by their other customers. The network load tends to be higher in the working hours than, e.g., at night. Thus, we like to find out if certain periodic patterns can be recognized which would allow us to identify, at which time of the day the connectivity parameters might not be sufficient to guarantee an effective AGV operation. To analyze and identify the existence of such periodic behavior, the testing script will run for a whole weekday, i.e., 1440 minutes.

VPN setup: The VPN software of choice for the cloud and remote communication patterns is OpenVPN, since it is a free and widely supported open-source Secure Sockets Layer (SSL)-based solution. The community edition of OpenVPN, however, has multiple features and versions that may influence the parameters of its connections. For instance, the newest version 2.4 offers a significantly enhanced behavior than the standard version 2.3 which is the default version pre-loaded for popular operative systems such as Ubuntu 16.04. In the following, we will refer to the OpenVPN setup parameters used in the various tests.

VI. Analysis of the Experimental Results

The experimental results of our tests are summarized in Tab. I. The Figs. 4, 5, and 6 depict the evolution of the latency and throughput over the test period for the city, cloud, and remote use cases discussed in Sect. IV. In addition, the figures include lines describing the linear regressions over the sensed latency and throughput values as well as the corresponding equations (see, e.g., [17]). While the coefficient of determination R^2 is very low, indicating that data does not fit the linear regression model well, the lines still provide hints about the generalized behaviors across the test periods. This study does not focus on obtaining a model from any of the variables under survey, but to present a general overview of the performance for long distance communication applied to the remote control of AGVs.

Comparing the average latency μ results of the three scenarios in Tab. I, a positive correlation with the geographical separation between the computation resources is visible. The

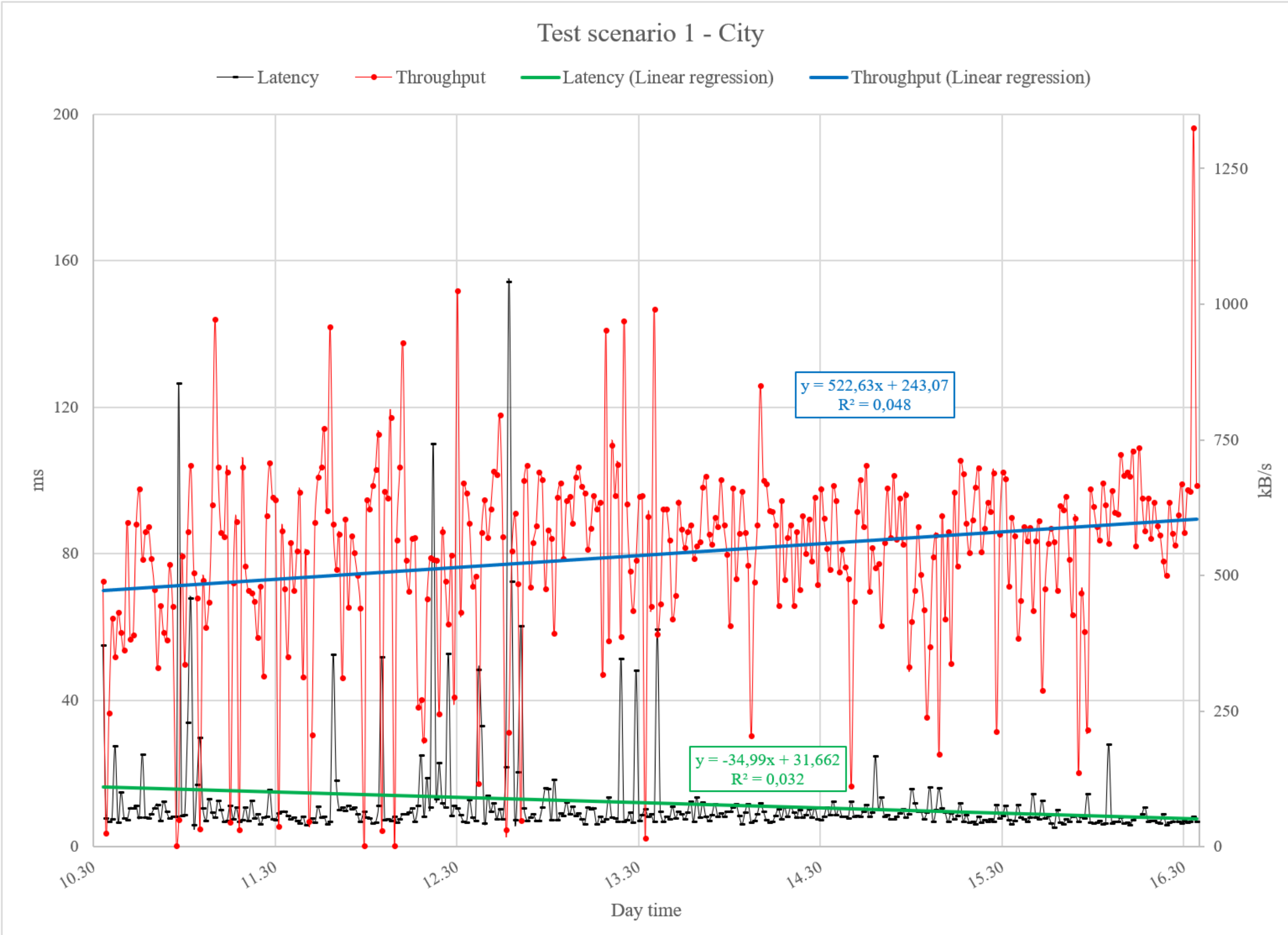

Fig. 4. Test results for scenario 1: City (24/03/2020). Computing resources in the Helsinki area (Finland)

latency results obtained for the city and cloud scenarios have a magnitude under 30 ms. This indicates appropriate responsiveness conditions facilitating a fluid communication for both communication patterns used. In contrast, the latency of around 360 ms for the remote scenario is considerably higher. This exactly matches the results of the latency tests between Melbourne and Trondheim [8] that were carried out in 2015 and discussed in Sect. II.

To evaluate the stability of the latency values, we should also consider the maxima, standard deviations, and variation coefficients. As indicated by the lowest coefficient of variation c_v and a relatively low standard deviation σ, the remote scenario has the most stable performance while the other scenarios contain higher variances. In particular, the city scenario depicted in Fig. 4 shows significant fluctuations over time. The spikes on the cloud scenario are smaller as indicated by the lowest standard deviation σ, while the remote communication pattern-based scenario shows higher oscillations.

The average values for the throughput measurements point to a negative correlation with respect to the distances between the computational resources. A reason for this is that passing a larger number of routers increases the likelihood to pass one that is subject to heavy traffic. This may result in the activation of certain automated methods for congestion control over TCP networks (see [18]) which may reduce the throughput. To look into that, we conducted a separate route tracing test to find out the number of routers that are passed by the data packets. As expected, the networking over longer distances comprised more routers: In the city scenario five hops were enough to reach the recipient while for the remote case 14 routers had to be passed. These test results reveal the complexity scale in Not Standalone (NSA) network components for the communication patterns. On the other side, the throughput in the remote scenario is more stable than in the city case as indicated by the values σ and c_v.

To compute the availability values, we used 20 kB/s as our throughput threshold in order to address the experiment described in [16] that was discussed in Sect. V. We see that both, the city and cloud scenarios pass this threshold most of the time proving a good availability of the communication services. Depending on the safety requirements of a particular remote AGV system control operation, this QoS may be

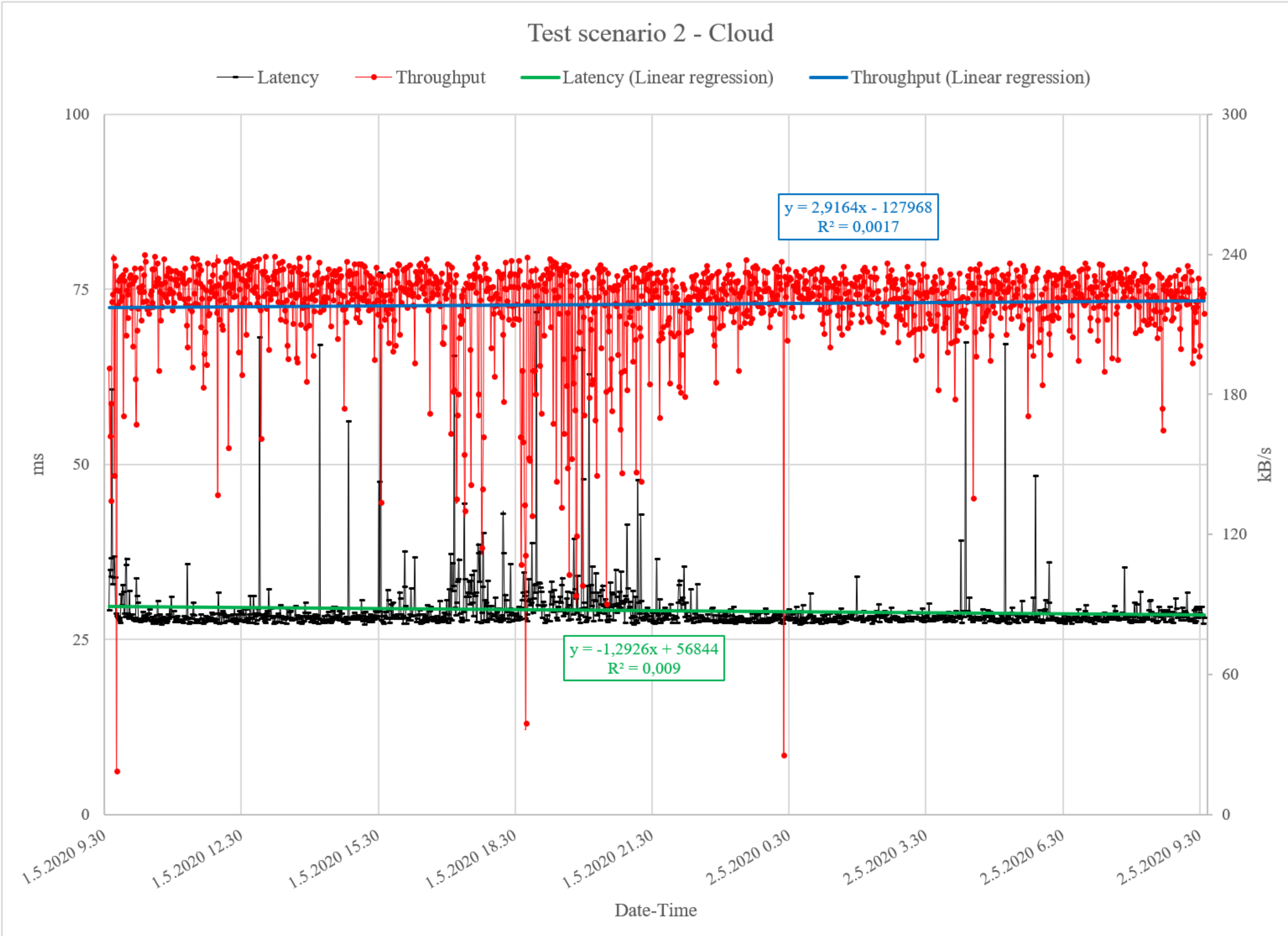

Fig. 5. Test results for scenario 2: Cloud. Computing resources at Microsoft Azure in Amsterdam / Netherlands

sufficient for the direct monitoring and control of devices from the control center. But the test also reveals that the long-distance connection used in the remote scenario renders a QoS that is insufficient or barely enough to allow us to control such systems between Espoo and Melbourne.

To assess the influence of the date and time on the tests, we considered the time zones at the geographical locations of the computing resources. While carrying out our tests, the time zones of Amsterdam, Helsinki and Melbourne were UTC+2, UTC+3, and UTC+10, respectively. This situation allowed us to experiment under conditions with similar time zones on both ends, i.e., Amsterdam and Helsinki, as well as a special situation, i.e., between Melbourne and Helsinki, where both terminals ran at very distinct times of the day.

The throughput behaviour of the similar time zone scenarios (city and cloud) presented relatively high instability that can be attributed to load-handling routines, performed automatically for one or several devices at the time in the network component. We could recognize a higher coefficient of variability c_v, represented by highly disperse spikes on Figs. 4 and 5, especially during working or leisure hours (late morning to midnight). In contrast, on late night, the spikes are closer to the mean value, making the throughput look more stable.

As shown by the low gradient of the linear regressions in Fig. 6, the remote scenario shows better stability. However, the figure also reveals easily recognizable patterns for both, latency and throughput tests. The regularity of the sinus-like oscillations in periods of four hours makes it difficult to attribute them to time of the day factors since their frequency seems to be too low for pointing at workload-based load-handling routines as observed in the other scenarios. While the connection is stable for the remote scenario, the absolute throughput values are significantly lower compared with the other scenarios. The results are too variable to guarantee a potentially appropriate interface between the computing resources and the AGVs.

The other two scenarios depict periods with accumulated throughput and latency spikes. In the city case, this phenomenon appears in the morning hours, while in the cloud case it happens in the evening. As mentioned above, this can be related to network load and time of the day factors, which might impact the shared communication media (e.g., daytime

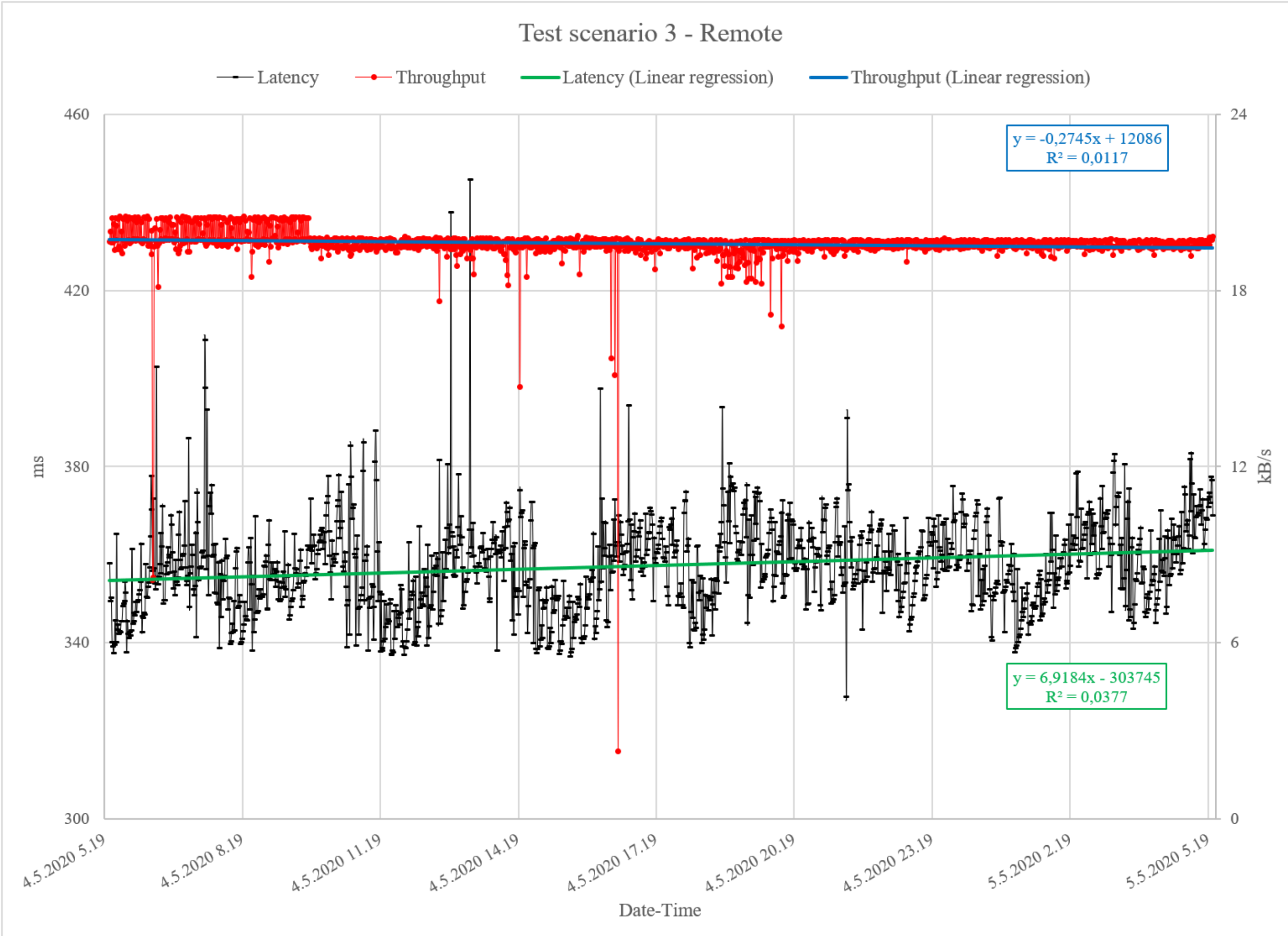

Fig. 6. Test results for scenario 3: Remote. Computing resources at RMIT's VXLab (Melbourne, Australia)

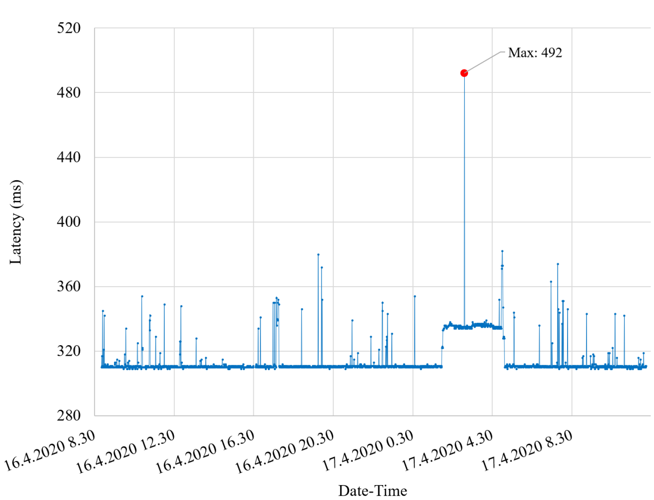

Fig. 7. Ping latency tests Aalto-RMIT

working or studying hours on weekdays as well as media streaming after working hours or during public holidays). Such adjacent activity can be monitored and the bandwidth share controlled locally, but it might be difficult to measure it independently and correlate it with the tests performed in this study. Nevertheless, in both scenarios the effects are not very significant and their throughput values nearly always remain above the availability threshold. Thus, the external disturbances concentrated in time lapses do not seem to have serious negative impacts on the system availability.

Finally, we want to look on the influence of the VPN used. As discussed, we applied OpenVPN that uses static keys with a size of 2048 bits, complying the Advanced Encryption Standard (AES) on the Cipher Block Chaining mode (CBC). The cryptographical hash in use is based on SHA-256 using a default parameter selection. The standard version of OpenVPN (version 2.3 for Ubuntu 16.04) was set to work initially on the connection-less transport protocol UDP. It could be successfully initialized for the use with TCP in the city and cloud scenarios, but did not work with the remote scenario. However, version 2.4 of OpenVPN proved to be a better fit for being used with TCP. Applying this version, we easily managed to establish the connections properly in all three scenarios. Therefore we used this version in all test runs.

Figure 7 depicts another latency experiment between Aalto and RMIT using ping. We are not sure about the reason for the slight deviation of the latency from 310 to 340 ms but assume some maintenance work in the global communication network. We mention this case to demonstrate that there can, indeed, be external reasons that are beyond the control of the AGV operator. In a practical use of remote control systems over long distances the appearance of such surprising effects should always be taken into consideration.

By comparing the results with the literature [5]–[7], discussed in Sect. II, we could recognize that our average latency results for the city and cloud scenarios are under the maximum delay expected on their experiments (95 ms). This reveals that it is possible to achieve a data transfer performance identical to the local communication pattern for other scenarios using the cloud or remote patterns. However, in the scenario linking Espoo with Melbourne, the remote communication pattern shows values of the relevant QoS parameters that are too deficient for the data exchange requirements in AGV control applications. The typical refresh rate for the control systems of 100 ms, would not be properly covered with latency values of around 357 ms. Nevertheless, the relatively good variability values in the remote scenario that are also confirmed by the data reported in [8], are a bright spot. While not for highly sensitive tasks demanding the fulfillment of hard real time conditions, long-distance control over continents may work for control scenarios demanding less strict latency and throughput parameters, e.g., to use remote monitoring for predictive maintenance.

VII. Conclusion

Using communication patterns for the remote control and simulation of AGVs, we explored connections between our European site in Espoo, Finland, and Melbourne, Australia as well as between Espoo and the Microsoft Azure cloud in Amsterdam. The gathering and interpretation of empirical data on the network performance was a key aspect of this paper. This data can serve as a first indication on the expected QoS parameters like latency and throughput in larger setups.

Currently, many industrial AGVs depend on stand alone WLAN solutions (as depicted in the local communication pattern). In the near future, this layout will be complemented and, perhaps, replaced by interconnected solutions, e.g., the Ultra-Reliable Low-Latency Communication (URLLC) technology in 5G. These novel remote control procedures will be an enhanced way to link AGVs to industrial automation platforms demanded by Industry 4.0. To safely realize the distributed control procedures, novel remote communication patterns are required. Further, new use cases to facilitate a better application of distributed computing as well as the use of digital twins and simulation resources need to be studied.

Future work will incorporate more devices such as UR 3 robots and PLCs or PLC-like devices in our facilities. To research the impact of 5G-technology on the connectivity, we further plan to utilize a new 5G-Lab that is currently established at NTNU in Trondheim. Moreover, larger applications including augmented and virtual reality are in our scope for future work as well.

References

[1] J. O. Blech, I. Peake, H. Schmidt, M. Kande, A. Rahman, S. Ramaswamy, S. D. Sudarsan, and V. Narayanan, "Efficient Incident Handling in Industrial Automation through Collaborative Engineering," in *20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2015, pp. 1–8.

[2] G. S. Martínez, T. Karhela, R. Ruusu, S. Sierla, and V. Vyatkin, "An Integrated Implementation Methodology of a Lifecycle-wide Tracking Simulation Architecture," *IEEE Access*, vol. 6, pp. 15 391–15 407, 2018.

[3] Aalto University, School of Electrical Engineering, "Factory of the Future," https://www.aalto.fi/en/futurefactory, accessed: 2020-05-08.

[4] I. Peake, J. O. Blech, L. Fernando, H. Schmidt, R. Sreenivasamurthy, and S. D. Sudarsan, "Visualization Facilities for Distributed and Remote Industrial Automation: VxLab," in *20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2015, pp. 1–4.

[5] C. Lozoya, P. Martí, M. Velasco, J. M. Fuertes, and E. X. Martín, "Simulation Study of a Remote Wireless Path Tracking Control with Delay Estimation for an Autonomous Guided Vehicle," *International Journal of Advanced Manufacturing Technology*, vol. 52, no. 5, pp. 751–761, 2011.

[6] M. de Ryck, M. Versteyhe, and F. Debrouwere, "Automated Guided Vehicle Systems, State-of-the-art Control Algorithms and Techniques," *Journal of Manufacturing Systems*, vol. 54, pp. 152–173, 2020.

[7] E. Cardarelli, V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Cooperative Cloud Robotics Architecture for the Coordination of Multi-AGV Systems in Industrial Warehouses," *Mechatronics*, vol. 45, pp. 1–13, 2017.

[8] P. Herrmann, A. Svae, H. H. Svendsen, and J. O. Blech, "Collaborative Model-based Development of a Remote Train Monitoring System," in *11th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*. Rome, Italy: SciTePress, 2016, pp. 383–390.

[9] S. Hordvik, K. Øseth, H. H. Svendsen, J. O. Blech, and P. Herrmann, "Model-based Engineering and Spatiotemporal Analysis of Transport Systems," in *Evaluation of Novel Approaches to Software Engineering*, ser. CCIS 703. Springer-Verlag, 2017, pp. 44–65.

[10] AMQP.org, "Advanced Message Queuing Protocol (AMQP)," www.amqp.org/, 2020, accessed: 2020-04-20.

[11] A. Svae, "Remote Monitoring of Lego-Mindstorm Trains," Project Thesis, Norwegian University of Science and Technology, Trondheim, 2016.

[12] J. O. Blech, I. D. Peake, G. Jahn, and R. Snooks, "A Software Platform for Architectural Robots," in *24th Australasian Software Engineering Conference (ASWEC)*, 2015, pp. 54–58.

[13] H. F. Atlam, A. Alenezi, A. Alharthi, R. J. Walters, and G. B. Wills, "Integration of Cloud Computing with Internet of Things: Challenges and Open Issues," in *IEEE International Conferences on Internet of Things (iThings), Green Computing and Communications (GreenCom), Cyber, Physical and Social Computing (CPSCom), and Smart Data (SmartData)*. IEEE, 2017, pp. 670–675.

[14] MQTT.org, "Message Queuing Telemetry Transport (MQTT)," mqtt.org/, accessed: 2020-05-07.

[15] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll, "OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols," in *IEEE International Conference on Industrial Technology (ICIT)*, 2019.

[16] A. Manzi, L. Fiorini, R. Limosani, P. Sincak, P. Dario, and F. Cavallo, "Use Case Evaluation of a Cloud Robotics Teleoperation System," in *5th IEEE International Conference on Cloud Networking (Cloudnet)*, 2016, pp. 208–211.

[17] K. P. Murphy, *Machine Learning — A Probabilistic Perspective*. MIT Press, 2012.

[18] "Congestion Control in IP/TCP Internetworks," RFC 896, Jan. 1984. [Online]. Available: https://rfc-editor.org/rfc/rfc896.txt